NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

**PARTICLE FILTERING METHODS FOR
INCORPORATING INTELLIGENCE UPDATES**

by

Jesse A. Nunez

March 2017

Thesis Advisor:	Dashi I. Singham
Second Reader:	Michael P. Atkinson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE March 2017	3. REPORT TYPE AND DATES COVERED Master's Thesis 09-01-2015 to 03-31-2017	
4. TITLE AND SUBTITLE PARTICLE FILTERING METHODS FOR INCORPORATING INTELLIGENCE UP-DATES			5. FUNDING NUMBERS
6. AUTHOR(S) Jesse A. Nunez			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Due to uncertainty in target locations, stochastic models are implemented to provide a representation of location distribution. The reliability of these models has a profound effect on the ability to successfully interdict these targets. A key factor in the reliability of a model is the incorporation of information updates. A common method for incorporating information updates is Kalman filtering. However, given the probable nonlinear and non-Gaussian nature of target movement models, the fidelity of solutions provided by Kalman filtering could be significantly degraded. A more robust methodology needs to be employed.

This thesis uses an updating algorithm known as particle filtering to incorporate information updates concerning the target's position. Particle filtering is a nonparametric filtering technique that is adaptable and flexible. The particle filter is incorporated into a model that uses a stochastic process known as a Brownian bridge to model target movement. A Brownian bridge models target movement with minimal information and allows for uncertainty during periods when target location is unknown. As new intelligence arrives, the particle filter is used to update a probabilistic heat map of target position. The main goal of this thesis is to design a stochastic model integrating both the Brownian bridge model and particle filtering.

14. SUBJECT TERMS Brownian bridge movement models, particle filter, stochastic modeling, nonlinear filtering, simulations	15. NUMBER OF PAGES 79
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

i

THIS PAGE INTENTIONALLY LEFT BLANK

PARTICLE FILTERING METHODS FOR INCORPORATING INTELLIGENCE UPDATES

Jesse A. Nunez
Lieutenant, United States Navy
B.S., North Carolina State University, 2005
M.S., Auburn University, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2017

Approved by: Dashi I. Singham
Thesis Advisor

Michael P. Atkinson
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Due to uncertainty in target locations, stochastic models are implemented to provide a representation of location distribution. The reliability of these models has a profound effect on the ability to successfully interdict these targets. A key factor in the reliability of a model is the incorporation of information updates. A common method for incorporating information updates is Kalman filtering. However, given the probable nonlinear and non-Gaussian nature of target movement models, the fidelity of solutions provided by Kalman filtering could be significantly degraded. A more robust methodology needs to be employed.

This thesis uses an updating algorithm known as particle filtering to incorporate information updates concerning the target's position. Particle filtering is a nonparametric filtering technique that is adaptable and flexible. The particle filter is incorporated into a model that uses a stochastic process known as a Brownian bridge to model target movement. A Brownian bridge models target movement with minimal information and allows for uncertainty during periods when target location is unknown. As new intelligence arrives, the particle filter is used to update a probabilistic heat map of target position. The main goal of this thesis is to design a stochastic model integrating both the Brownian bridge model and particle filtering.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AOR   Area of Responsibility

BBMM   Brownian Bridge Movement Model

CLT   Central Limit Theorem

CMIS   Center for Multi-Intelligence Studies

GPS   Global Positioning System

KF   Kalman Filter

MATLAB   Matrix Laboratory Program

MOE   Measure of Effectiveness

MPRA   Maritime Patrol and Reconnaissance Aircraft

PDF   Probability Density Function

PF   Particle Filter

SIS   Sequential Importance Sampling

SBBM   Simulated Brownian Bridge Model

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Stochastic models, particularly ones of a time series nature, must be dynamic in order to represent changing conditions. These models often require a mechanism for updating new information. For modeling target movement, information updates that affect the uncertainty surrounding target locations are important. Inferior or invalid information updating techniques can render an otherwise effective stochastic model ineffective. This thesis addresses the problem of how information gets incorporated into a stochastic model. Specifically, the focus is to incorporate particle filtering techniques into a model for target location and tracking based on Brownian bridges. The scenario chosen for this thesis is a counter-narcotics smuggling scenario in the Eastern Pacific Ocean.

Target tracking is a well-researched area of stochastic modeling. There is a variety of modeling methods such as epi-spline approximation (Campos 2014) and simulation of Brownian bridges for target movement (Cheng 2016). A Brownian bridge is a continuous time stochastic process defined as Brownian motion fixed at two end points and can be used to model target motion. The model in this thesis, introduced in Cheng (2016), attempts to estimate the distribution of the target's location through the aggregation of numerous simulated Brownian bridges of the target path according to intelligence information. This model creates a probabilistic heat map based on the simulated paths, i.e., a location with a higher concentration of simulated paths corresponds to a higher probability that the target is at that location. Intelligence sensors are placed along the target path where information updates determine whether a target is detected. With the stochastic model selected, the problem becomes selecting the appropriate information-updating mechanism.

There exists a variety of ways to incorporate information updating. These methods include simple heuristics, such as a pass/fail acceptance test, and advanced filtering techniques, such as Kalman filtering and particle filtering. One of the more prevalent techniques is Kalman filtering. Kalman filtering attempts to predict the current and future states of a model using past and current observations. This technique relies on the assumption of linear state and observation equations as well as a normally distributed error, or noise. These assumptions allow for low computational run times and analytical solutions. However, if these assumptions prove to be invalid, the Kalman filter has the propensity to yield

questionable results, which can lead to poor estimates of the state variable distribution. Particle filters have more flexible methods for updating the state variables and hence are used for incorporating sensor updates to target locations in this thesis.

The particle filter, introduced in Gordon et al. (1993), is a nonparametric filter that does not rely on the assumptions of linearity and normality. Particle filters generate samples of the underlying state-variable (in this case, target location) distribution to estimate the current and future state of the model. As new information is introduced, each particle is assigned a weight and then a resampling with replacement is conducted with respect to those weights. These new particles are then iterated forward to the next time step. Issues can arise when using a particle filter. Particle degeneracy (the propensity of the resampled particles to collapse to a single unique particle) and the curse of dimensionality (computational complexity increases exponentially in response to increases in the dimension of the state space) are the most prevalent issues. Mitigation techniques need to be incorporated in order to use a particle filter effectively.

This thesis implements particle filtering into the Brownian bridge model for target tracking. The structure of the model is ideal for the integration of a particle filter. Brownian bridge paths are now represented as particles. Intelligence sensors placed throughout the path of the target act as surrogates for information. As a particle passes through the sensor, it is weighted based on a sensor probability of detection, particle position in relation to the sensor, and a Boolean detection flag, which is determined by whether the sensor detected the target. These weighted particles are resampled only when the sensor is active demonstrating adaptive sampling, which is a mitigation technique used against particle degeneracy and the curse of dimensionality. Adaptive resampling is a technique that limits particle filtering to times when a certain criteria is met. In order to mitigate particle degeneracy, the particles are roughened prior to advancing to the next iteration. This thesis introduces a particle roughening technique that is exclusive to a Brownian bridge model for target motion. This technique maintains the current position of the particle while resampling by creating a new Brownian bridge path. Figure 1 shows snapshots from a Brownian bridge model that has been enhanced with particle filtering.
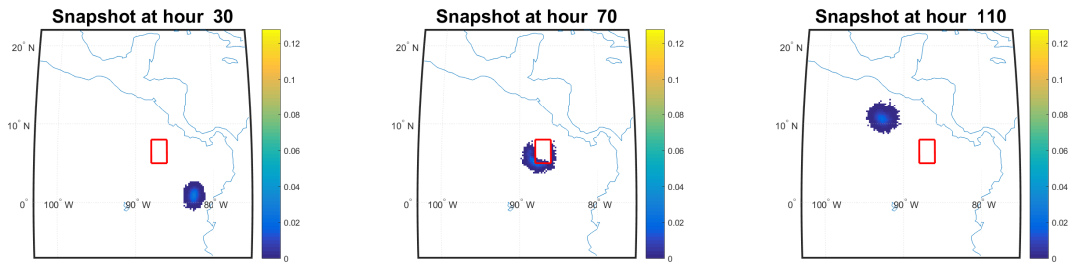
Figure 1. Brownian Bridge Movement Model Incorporating Particle Filter Updates in a South American Counter-Narcotics Scenario.

The goal of this thesis is to demonstrate that a robust information updating algorithm can be incorporated into a Brownian bridge model for target motion. This is accomplished through experimentation where the objective is to provide a realistic model. With the fully developed model, this research describes the experimentation conducted in order to refine the methodologies developed. Experiments concerning realistic sensor size, alternative weighting schemes, and degeneracy mitigation techniques are performed. This thesis successfully incorporates a robust particle filtering algorithm into a highly adaptable Brownian bridge model and demonstrates the versatility of these advanced methods.

References:

Campos S (2014) An analysis of mathematical models to improve counterdrug smuggling operations. Master's thesis, Naval Postgraduate School.

Cheng CC (2016) A Brownian bridge movement model to track mobile targets. Master's thesis, Naval Postgraduate School.

Gordon NJ, Salmond DJ, Smith AF (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* 140(2):107– 113.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I offer my sincerest thanks to Professor Dashi Singham for graciously serving as my thesis advisor. The guidance, advice, and direction that she provided me throughout the process is much appreciated.

I also thank Professor Michael Atkinson for his willingness to be my second reader. His critiques and recommendations elevated the quality of this work.

I am especially thankful for my wife, Lisa, whose daily encouragement and support make me strive to be the best that I can. I could not have done this without her. Thank you to my family for always being there for me.

Lastly, I thank the Naval Postgraduate School Department of Operations Research for providing me with the necessary tools and support to allow this goal to become a realization.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

An important element of a stochastic model is its ability to incorporate changes in the information that drives the model. Many different methods for implementing information updates are available, each with its own strengths and weaknesses. The purpose of this thesis is to show how to use particle filtering methods to incorporate information updates of a stochastic target movement model, specifically a Brownian bridge model for target motion. This thesis provides a discussion on the Brownian bridge movement model (BBMM), which motivates our simulation model, particle filtering, and the joint implementation into a realistic model for tracking the distribution of a target's location.

1.1 Overview of the Model

The overall problem that this thesis addresses is the problem of target tracking. Target tracking problems encompass a variety of situations, from map-following in a vehicle Global Positioning System (GPS) to target location estimation. The problem that is specifically addressed is the estimation of a target's location. For this problem, there are two requirements: a stochastic model for target movement and an updating method to incorporate new information.

1.1.1 Stochastic Model

The level of uncertainty present in a target location estimation problem gives credence to using a stochastic model. While there are many categories of stochastic models available, the one that is most relevant for our purposes is the Brownian bridge movement model. This model for target motion was first suggested by Bullard (1999) due to its unique characteristics, which include "fixing" the target path to specific endpoints. Extensive work in animal migratory movements, such as tracking black bears (Horne et al. 2007), has further refined this method. Efforts have also been conducted to "develop a framework for algorithmic movement analysis using the Brownian bridge movement model" (Buchin et al. 2012).

The underlying continuous time stochastic process that drives the BBMM is the Brownian motion. A Brownian bridge is Brownian motion conditional on being fixed at two endpoints. While the BBMM relies on the analytical distribution of Brownian bridges between two endpoints, the model employed by this thesis aggregates numerous simulated Brownian bridges between two areas in order to create a heat map of target locations at specified time steps. This model is referred to as the Simulated Brownian Bridge Model (SBBM). This two-dimensional heat map serves as a representation of the target location probability density at a given time interval. With a stochastic model selected, the next objective is to determine a methodology for incorporating information updates.

1.1.2   Information Updating Method

The mechanism for incorporating information updates is an important component of a stochastic model. There is a plethora of options when it comes to updating stochastic information. One of the most popular methods used today is the Kalman filter (KF) introduced in Kalman (1960). Another more recent method introduced in Gordon, Salmond, and Smith (1993) is the particle filter (PF).

Kalman filters are an easily implementable and computationally inexpensive method for updating information with uses in navigation, robotics, and even neuroscience (Wolpert and Ghahramani 2000). The KF makes a prediction of the state space from a given observation and then updates its state space estimation as more information is incorporated. Its ease of use and analytical intuition stems from its assumptions of a linear state equation and normal errors. Generalizations and extensions to the KF, such as the extended Kalman filter and unscented Kalman filter, have been developed to combat nonlinearities. However, these extensions merely linearize a nonlinear system that has the potential to yield inaccurate estimations.

There are strong underlying assumptions that are present in the KF that can lead to question-able results in situations where strong nonlinearity and/or non-normality are present. Both of these issues can arise with target tracking, so a more robust method for incorporating information updates needs to be used. The updating method that is the focus of this thesis is the particle filter.

A particle filter is a nonparametric method for incorporating information updates. The

strength of the PF lies in its simplicity. The PF uses Monte Carlo sampling of a probability density to create particles, which are sampled over time to estimate the distribution of the underlying state space. A particle goes through a weighting algorithm as it progresses through time, where the weight is assigned in conjunction with an information update at a given iteration. A resampling with replacement occurs based on the weight of the particle. Particles with a higher weight have a higher propensity of being sampled and carried to the next iteration. The PF has the capability to incorporate additional embellishments that can be tailored to a specific problem. An important property of the PF is that there are no major underlying assumptions needed for the state-transition equations of the underlying state space, making it a flexible method of information updating. Numerous applications implement particle filtering techniques, such as car positioning by map matching, car positioning by radio frequency measurements (cellular towers), aircraft positioning by map matching or terrain navigation, integrated navigation, target tracking and car collision avoidance (Gustafsson et al. 2002, Gustafsson 2010). The PF is also used in physical models such as thermal transfer (Orlande et al. 2011), where it fares significantly better in its estimation than a Kalman filter (due to the nonlinearity present in the thermal transfer experiments).

1.2 Model Scenario

The scenario that is used in the simulated Brownian bridge model with particle filter updates is maritime narcotics trafficking in the Eastern Pacific Ocean near South and Central America. This scenario is easily implemented as a SBBM. The PF updates the model with information received by intelligence gathered from an active sensor.

1.2.1 Background

Estimates show that approximately 90% of the illegal narcotics entering the United States arrive via South and Central America (Dudley 2010). Figure 1.1 highlights the major drug trafficking routes out of South and Central America. As individual countries along the Central American Isthmus increase their counter-narcotics efforts along land-based routes, narcotics traffickers must rely on maritime approaches to successfully transport their illicit cargo.

Figure 1.1. Drug Trafficking Routes from Central and South America. Source: Kelly (2012).

The United States government has invested significant capital in attempts to thwart transnational drug smuggling efforts. However, in the current economic state, with tightening fiscal constraints, the old system of "throwing money at the problem" is no longer feasible. Since more money is not to be expected, efficient use of current operating budgets is the only real option. Smarter modeling of narcotic traffickers' smuggling patterns is important.

Modeling narcotic smuggling patterns is a well-researched topic. Different methodologies have been incorporated in the search for more accurate models. One of the main differences in these methods is the way that the probability density function (PDF) of the target location is estimated. Examples include epi-spline approximation (Campos 2014) or Brownian bridge modeling (Cheng 2016). Figure 1.2 displays the historical trends of South American narcotics smuggling and visually suggests that a Brownian bridge model is an appropriate model due to the high variability around the center of the paths.

4
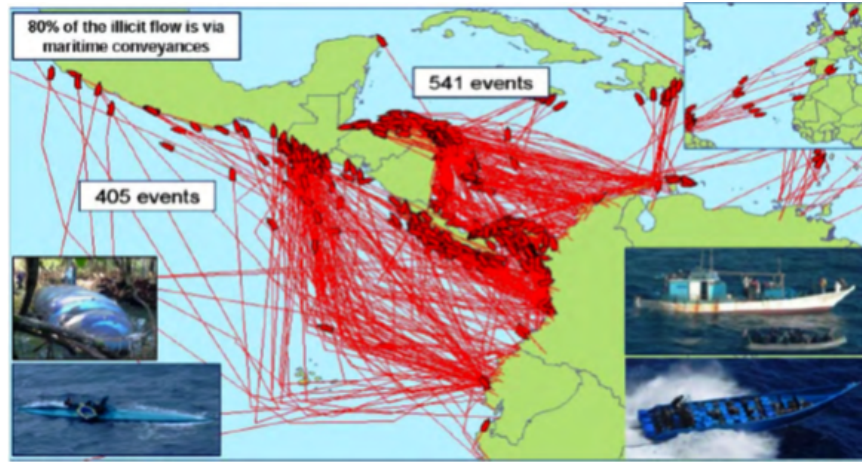
Figure 1.2. Maritime Drug Transit through the Caribbean and Central America. Source: United States Senate Caucus on International Narcotics Control (2012).

In the counter-narcotics scenario, information updating is achieved through intelligence. Sensors are placed along the projected target path where a detection area is believed to operate with a corresponding probability of detection. In this thesis, these intelligence updates are incorporated via the PF. The Brownian bridge paths created by the SBBM serve as the particles to the PF. As the target probability density particles (displayed as a probabilistic heat map) pass through the active sensor region, the PF updates the model given a positive or negative intelligence update. The longer a sensor goes without seeing a target, the less likely that the particles within that sensor region are resampled for future time steps.

1.2.2 Details

The start and end point of our scenario will be the Manabi Province of Ecuador and Parque Nacional Lagunas de Chacahua in Oaxaca, Mexico, respectively. There are three typical smuggling routes in the Eastern Pacific from South America (Ecuador or Columbia) to the Central American Isthmus (Mexico or Guatemala). The first path tends to hug the coastline just outside of territorial waters, primarily chosen to blend in with legitimate fishing traffic. The second route option causes smugglers to travel out west then turn back in an effort to avoid counter-narcotic patrol areas. The third path uses the most direct course from the start to end locations, which minimizes the time spent in transit. For our model, the third option is

chosen for its mathematical and computational simplicity, although any scenario is feasible with our chosen methods. Additionally, the variability in the model can be increased to simulate more extreme paths. Sensors will be placed along the projected target path in our experiments, though ongoing work is developing new methods for sensor placement.

1.2.3 Model Assumptions and Simplifications

Here is the list of assumptions and simplifications to the model:

- The target does not react to sensor presence.
- Sensors must be disjoint in time and space.
- Sensors are assumed to have a bounded coverage area, with a constant probability of detection within their coverage area.

Additional assumptions that address the specific type of sensor platform and target class are addressed in Chapter 4.

1.3 Research Questions

With the simulated Brownian bridge model designated as the target motion estimation model, this research can focus on intelligence updating methods, specifically particle filtering. The intent of this thesis is to show how particle filtering can be used to incorporate those intelligence updates into the SBBM and to show how particle filtering can be used to update a target's position distribution based on sensor information. This thesis will attempt to answer the following questions:

1. What type of fidelity can be achieved by using particle filtering that cannot be achieved using Gaussian methods, such as Kalman filtering?
2. Why would a particle filter be uniquely beneficial to the target tracking problem using the SBBM?
3. Are there any embellishments that can be made to the particle filtering method to reduce its computational complexity or increase its resolution?
4. How can the methodologies or algorithms developed in this thesis be used on a large scale in practical applications, i.e., multiple targets, multiple paths, negative intelligence?

1.4 Benefit of Thesis

The main benefit of this thesis is to provide a more robust methodology for incorporating intelligence updates into a stochastic model for target tracking. Due to the non-parametric assumptions of the PF, flexible methods for incorporating sensor updates can be developed. This can lead to a better method for estimating and updating the distribution of a target's location by making fewer assumptions on the targets behavior and sensor information.

1.5 Methodology

This thesis provides an in-depth overview of the literature on the Brownian bridge movement model and particle filter methods. Chapter 2 is an overview of the BBMM discussing the foundational mathematics beneath its creation, and recent developments using a simulated Brownian bridge model. Chapter 3 details the discussion of PF development with corresponding mathematics; explains common issues with PF as well as corresponding remedies; and introduces the particular PF algorithm that is developed for this thesis. Finally, the merger of SBBM and PF is presented along with corresponding experimentation in Chapter 4. The thesis is concluded in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Constructing the Simulated Brownian Bridge Model

The stochastic model that this thesis employs for target movement is a simulated Brownian bridge model, based on the Brownian bridge movement model. The BBMM is commonly used to represent animal migratory patterns and movement since its introduction by Bullard (1999). The BBMM assumes Brownian bridge motion between a start and end point, and constructs an analytical model for the distribution of an animal's location based on properties of Brownian bridges. The rationale for using this model for animal movement patterns is that animals do not operate in completely random patterns, but rather have a tendency toward their "home range." In addition to home range estimates, additional applications of the BBMM have been proposed such as estimating migration routes and resource selection (Horne et al. 2007). The BBMM is also used in the analysis of low resolution trajectory data because it "assumes random movement between sample points" (Buchin et al. 2012). The SBBM simulates Brownian bridge paths between a start and end point, and aggregates those paths to estimate the distribution of a target's location. The main advantage of the SBBM is that additional complexity (updates, randomized start and end points, etc.) can be added to the model to influence the simulated Brownian bridge paths easily. It would be difficult to maintain analytical estimates of target location using the BBMM while introducing these features. There have been recent attempts to model target motion in the South China Sea using the SBBM (Cheng 2016). An extension of the model used by Cheng is the driving force behind the algorithms discussed in later sections. This section will provide a brief background on the SBBM, which relies on simulated Brownian motion and Brownian bridges.

2.1   Random Walk

The most basic principle underlying a Brownian motion is a "random walk." In mathematics, a random walk is a sequence defined by a cumulative sum of independent and identically distributed random variables. The random walk, a discrete stochastic sequence, S_n, can be

expressed by the following formula:

$$S_n = \sum_{i=1}^{n} X_i, \tag{2.1}$$

where X_i is a random variable and n is the number of terms in the sequence. A nice feature of the random walk process is that there is no distributional requirement placed on X_i. However, since the central limit theorem (CLT) precludes the use of random variables with infinite variance, a similar restriction is placed on random walks that simulate Brownian motion approximations.

2.2 Brownian Motion

The transition from a random walk process to Brownian motion is a transition from a discrete-time stochastic process to a continuous-time stochastic process. In (2.1), S_n is created by a sequence of random variables X_i with a specified mean μ and variance σ^2. According to the CLT, under certain assumptions,

$$\lim_{n \to \infty} \frac{S_n}{\sqrt{n}} \to N(\mu, \sigma^2). \tag{2.2}$$

Assume that the random walk is a simple symmetric random walk that is defined when X_i is from a binomial distribution with one trial and with equal probability of drawing a -1 or 1 (Alm 2002). The limiting distribution in this case is a standard normal random variable according to the CLT. A continuous-time stochastic process that is the scaled limit of this discrete-time process is defined such that:

$$W_n(t) = \frac{S_{\lfloor nt \rfloor}}{\sqrt{n}}, \tag{2.3}$$

where $t \in (0, \infty)$ and $\lfloor nt \rfloor$ is the largest integer less than or equal to nt. A property of (2.3) is that $W_n(t)$ converges in distribution $W(t)$ as $n \to \infty$, where $W(t)$ is Brownian motion (Lalley 2013).

There are four key properties of Brownian motion:

1. $W(0) = 0$.

10

2. $W(t)$ is continuous for all t.
3. $W(t)$ is normally distributed with a mean of 0 and a variance of $\sigma^2 t$.
4. $W(t)$ exhibits stationary and independent increments.

Additionally, the covariance structure is determined as follows:

$$Cov\left(W(s), W(t)\right) = \sigma^2 \min(s, t). \tag{2.4}$$

Brownian motion provides the foundation required for the Brownian bridge.

2.3   Brownian Bridge

The BBMM and SBBM rely on the Brownian bridge, which is derived from Brownian motion. Simply stated, a Brownian bridge is Brownian motion tied to two specified endpoints, one at time 0 and one at another chosen time. Denote $B(t)$ as the value of a Brownian bridge at time t. In the standard Brownian bridge, the time horizon is fixed at $t \in [0, 1]$. In order to fix an endpoint, the additional property $B(1) = 0$ is introduced. Let $W(t)$ be a standard Brownian motion with mean $\mu = 0$ and variance $\sigma^2 = 1$. The Brownian bridge is represented by the following equation:

$$B(t) = W(t) - tW(1), \; t \in [0, 1]. \tag{2.5}$$

The properties of a standard Brownian bridge are:

1. $B(0) = 0$.
2. $B(1) = 0$.
3. $B(t)$ is continuous for all t.
4. $B(t)$ is normally distributed with a mean of 0 and a variance of $\sigma^2 t(1 - t)$, where $\sigma^2 = 1$ denotes a standard Brownian bridge.

The proof for property four is relatively straightforward (Siegrist 2015). Additionally, the covariance structure between two time points s and t is as follows:

$$Cov\left(B(t), B(s)\right) = \sigma^2[\min(s, t) - st]. \tag{2.6}$$

A discrete-time approximation of a Brownian bridge can be simulated based on a simulated

random walk, and an example with 10,000 time steps is given in Figure 2.1.
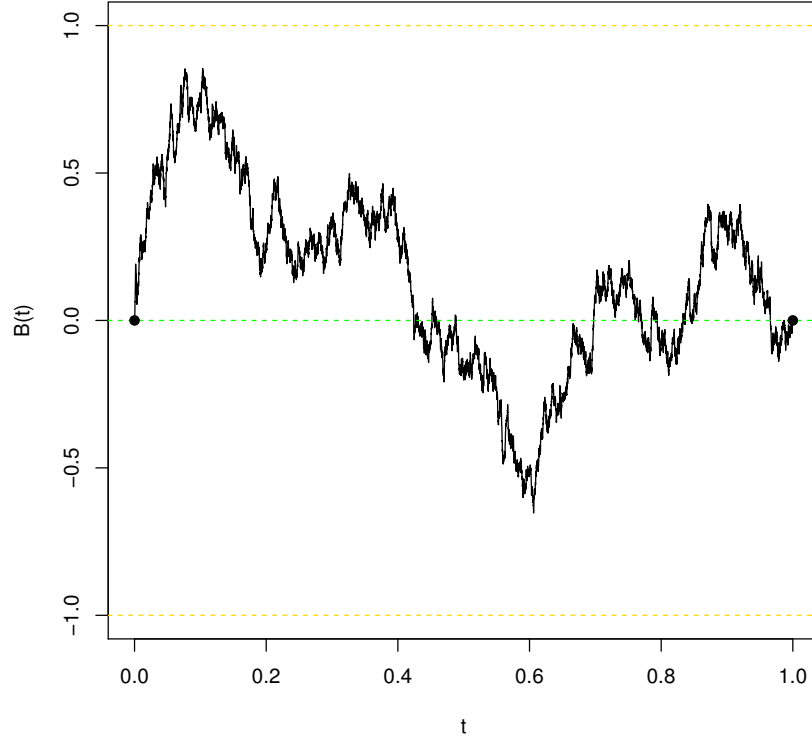
Figure 2.1. Simulated Approximation of a Standard Brownian Bridge with 10,000 Time Steps.

The Brownian bridge has the highest variance in the middle of the $[0, 1]$ range with the variance reducing to zero as time approaches either endpoint. The one-dimensional Brownian bridge is easily extended to higher dimensions, and this thesis focuses on the two-dimensional Brownian bridge, which forms the foundation of the SBBM. The two-dimensional Brownian bridge relies on two separate one-dimensional Brownian bridges:

$$
\begin{aligned}
B_x(t) &= W_x(t) - tW_x(1), \\
B_y(t) &= W_y(t) - tW_y(1),
\end{aligned}
$$

(2.7)

where W_x and B_x are the Brownian motion process and Brownian bridge, respectively, used

to generate the *x*-axis coordinates. The same idea applies for the *y*-axis in the second line of (2.7), which gives an independently generated second Brownian bridge. These equations can be generalized to allow for any starting and ending values for $B(0)$ and $B(1)$ in each axis. As an example, this process is used to simulate a discrete-time approximation to a Brownian bridge starting at (0,0) and ending at (1,1) in Figure 2.2.
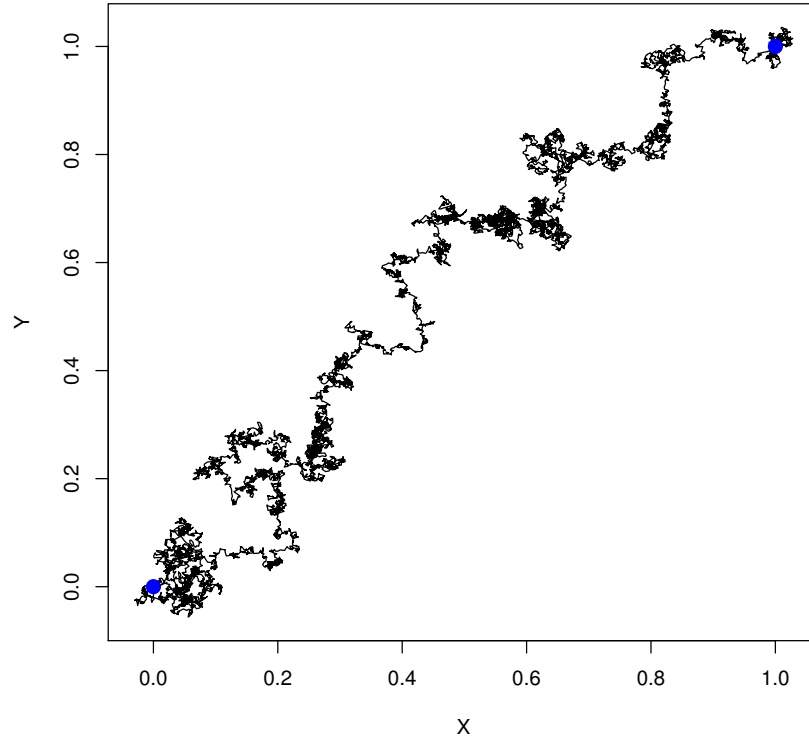
Figure 2.2. A Simulated Approximation of a Two-Dimensional Brownian Bridge.

2.4 The Simulated Brownian Bridge Model

The discrete-time simulated two-dimensional process serves as the foundation for the SBBM that assumes simulated Brownian bridge motion for the target between the start and end locations. Uncertainty in the start and end locations is incorporated by sampling the start and end points of each simulated Brownian bridge. At each time step, the locations of the

13

simulated paths can be aggregated into a two-dimensional heat map. The estimation of the distribution of the target's location using the heat map is essential to the development of the particle filter discussed in Chapter 3.

Horne et al. (2007) describes the Brownian bridge movement model as "a continuous-time stochastic model of movement in which the probability of being in an area is conditioned on starting and ending locations." In order to estimate the distribution of the target's location, a spatial heat map is employed. Using the methodology introduced by Bullard (1999) for tracking animal movements, a Brownian bridge movement model facilitates the estimation of the probability density heat map for target tracking. The model can be justified using the Brownian bridge properties in Section 2.3. Properties one and two are satisfied since the target's start and end location is assumed given with some uncertainty. Property three is obvious as the target must move in a continuous path from its start point to end point. Conceptually, property four is more difficult to justify. Weather and navigational error have elements of randomness associated with them. The Brownian bridge can employ the variance σ^2 as a tuning parameter to model random behavior. The variance function for a Brownian bridge over time is reasonable because a lower variability is expected when the target nears its endpoints. The normality assumption can be justified using the central limit theorem, though it is acknowledged that with simulated Brownian bridges a small time step would be needed for the distribution to be approximately normally distributed. However, even if the normality assumption proves invalid, detailed knowledge of a target's movements would be needed to fit a different distribution.

The SBBM simulates Brownian bridges based on user inputs through the following process. An initial time step vector is generated according to the number of time steps defined by the user. The start and end points of each Brownian bridge are sampled from two-dimensional starting and ending areas that represent the uncertainty in the start and end points. Then a random walk process is generated in the x and y dimensions to simulate Brownian motion. Using discretized versions of (2.7), the Brownian motion processes are converted to Brownian bridges. The multiple Brownian bridges are then aggregated at each time step to generate a probabilistic heat map. If more Brownian bridges appear in a given area at a given time, the heat maps will be "hotter" in that area, meaning there is a higher estimated probability the target is present in that location.

CHAPTER 3:
Particle Filter

With the simulated Brownian bridge model designated as the stochastic model for this thesis, the next step is to determine the method for updating information. Several options are available to analysts, including variants of a Kalman filter, as well as nonparametric filters such as a particle filter. A comparison is required to ascertain which one is best suited for the target tracking problem. This section discusses the reasons the particle filter was chosen and develops the particular algorithm applied to the SBBM.

3.1   Parametric Filters

The difficulty facing any information updating algorithm is the ability to discern signal from noise. The source of the difficulty is that an observation of the state space is often a function of the underlying state space, and the state space itself cannot be directly observed. Additionally, both the state variables and the observation have independent errors, or noise, associated with them. One of the more popular methods for extracting information about the state space from noisy observations is filtering. Filtering methods are used in a variety of applications such as navigation systems and robotics. For instance, in navigation, high precision GPS may be prohibitively expensive in commercially available hardware. However, low precision systems can be augmented with a filtering system (vehicle positioning through map matching) that provides the same level of fidelity with significantly lower costs (Gustafsson et al. 2002). In this vehicle positioning example, the state variable is the actual vehicle position while the observation is the perceived vehicle position from the low resolution GPS.

3.1.1   Kalman Filter

Analysts typically have many different types of filters at their disposal, but the most common is a Kalman filter. Introduced in Kalman (1960), the KF is a two-phase process of prediction and updating with the goal of predicting the current and future state through past and current observations. Once new data is incorporated into a filtering model, a new estimate of the state variable distribution is given. A KF can be easy to implement and computationally

15

inexpensive. The notation used to describe the KF in Boyd (2009) is employed in this thesis. The state variable vector at iteration t is called x_t. This state variable is a function of the state variable at the previous time step through a transition matrix defined as F and a white noise multivariate zero mean random variable ε_t. The noise variable ε_t is drawn independently from a multivariate normal distribution with a mean of zero and a covariance matrix E.

The state space relationship is defined as follows:

$$x_t = F x_{t-1} + \varepsilon_{t-1}. \tag{3.1}$$

The observation vector y_t is related to the state variable through a measurement matrix H and a separate noise variable. This noise variable γ_t, which could be considered measurement noise, is a multivariate normal random variable with a mean of zero and a covariance matrix Γ. Thus y_t is written as follows:

$$y_t = H x_t + \gamma_t. \tag{3.2}$$

The underlying assumptions of a KF are normally distributed noise variables and linear state and observation transition equations as in (3.1) and (3.2). Let $\tilde{x}_{t|t}$ and $\tilde{x}_{t+1|t}$ be the predictions of the current state and future state conditioned on the current observation y_t, respectively. Also, let $\Sigma_{t|t}$ and $\Sigma_{t+1|t}$ be the covariance matrices of the current and future state predictions $\tilde{x}_t$ and $\tilde{x}_{t+1}$, respectively, conditioned on the current observation y_t. The current state prediction at t and its corresponding variance given all information up to time t is expressed as (Boyd 2009):

$$
\begin{aligned}
\tilde{x}_{t|t} &= \tilde{x}_{t|t-1} + \Sigma_{t|t-1} H^T \left(H \Sigma_{t|t-1} H^T + \Gamma \right)^{-1} (y_t - H \tilde{x}_{t|t-1}), \\
\Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{t|t-1} H^T \left(H \Sigma_{t|t-1} H^T + \Gamma \right)^{-1} H \Sigma_{t|t-1}.
\end{aligned}
\tag{3.3}
$$

This provides the prediction of x_t as well as the variance of the prediction of x_t. In order to get the prediction of the future state x_{t+1}, (3.1) is conditioned on all available observations as:

$$x_{t+1}|y_{0:t} = F x_t|y_{0:t} + \varepsilon_t, \tag{3.4}$$

where $y_{0:t}$ are the set of observations up to time t. Using (3.3), the future state prediction

equation is as follows:

$$\tilde{x}_{t+1|t} = F\tilde{x}_{t|t}. \tag{3.5}$$

Consequently, using (3.5) yields the covariance matrix of the future prediction where $\Sigma_{t+1|t} = \mathrm{E}\left((\tilde{x}_{t+1|t} - x_{t+1})(\tilde{x}_{t+1|t} - x_{t+1})^T\right)$, where E is an expectation, and results in the following:

$$\Sigma_{t+1|t} = F\Sigma_{t|t}F^T + E. \tag{3.6}$$

Equations (3.3), (3.5), and (3.6) provide the means to estimate predicted values of the state variable recursively as data arrives. It should be noted that the initial values of the state variable and covariance matrix at time zero must be initialized based on user estimates.

While otherwise exceptionally useful, one drawback is that Kalman filtering can yield highly questionable results when there are nonlinearities present, incomplete information, or near singularity of the covariance matrix. Given the level of computing power available now compared to when Kalman created his algorithm nearly six decades ago, other filtering methods have been proposed to overcome these challenges.

3.1.2 Extension to the Kalman Filter

The first attempt to deal with nonlinear or non-Gaussian applications is to use a linear filter on the linear approximation of that application. An extended Kalman filter simply approximates all functions with a first-order Taylor series at the current estimate (Daum 2005). If only slight nonlinearities are present, an extended Kalman filter can sometimes provide the same fidelity as a KF at the same computational cost. As the extent of the nonlinearity increases, however, more advanced techniques are necessary.

The next evolution in nonlinear filters is the unscented Kalman filter. Like the extended Kalman filter, this method uses a linear approximation, but it implements a more advanced approximation technique "similar to the Gauss-Hermite quadrature for numerical approximation of multidimensional integrals" (Daum 2005). This approximation produces better results than the extended Kalman filter at roughly the same computational cost. However, the unscented Kalman filter is still based on a linear approximation of the state transition variables and the level of nonlinearity of the function it estimates will determine its accuracy. In order to overcome potential deficiencies in predictions associated with non-linearity

more robust filtering methods are required.

3.2 Foundations of Particle Filtering

Because of recent increases in computational power, Monte Carlo methods for nonparametric filtering have become more prevalent. Rather than attempting to approximate nonlinear functions, these methods sample from an estimate of the distribution of the state variable, and the resultant samples are called particles. Sequential Monte Carlo Methods attempt to "sample sequentially from a sequence of target probability densities" (Doucet and Johansen 2009). The two main issues with this approach are that the complexities of the probability densities make it difficult to sample, and sampling becomes computationally expensive as the iterations progress. These are defined as the complexity issue and computational issue, respectively. Importance sampling attempts to solve the complexity issue by employing an importance density function. This function is easier to sample from than the original distribution, and yields an importance weight for each particle that is used to determine the probability of future resampling for that particle. The computational problem can be mitigated through sequential importance sampling (SIS). Rather than sampling from the importance density of the entire state space in each iteration, which can have a high computational expense, the samples are drawn from the importance density at the current iteration conditioned on the particles from the previous time step (Minor 2011). In this case, the computational expense does not increase over time as future samples are generated from the particles at past time steps.

3.2.1 Particle Filtering through Bayesian Bootstrap Sampling

Although SIS helps resolve the computational and complexity issues addressed in the previous section, a new issue called particle degeneracy can arise. Particle degeneracy is the propensity of the set of particles to eventually collapse to one unique non-zero weighted particle as resampling continues from the same set of particles at each time step. After each iteration t, a weight of zero may be assigned to some particles through importance sampling. This zero weight leads to the "death" of those particles since there is zero probability that those particles will be sampled at the next iteration. The result is that only one unique particle with a positive weight remains after a certain number of iterations. A method for mitigating particle degeneracy was introduced by Gordon et al. (1993). In this paper, the

18

authors also present an alternative to the Kalman filter that does not require linear state transition functions and normal distributions for noise variables. This insight was called the Bayesian bootstrap filter, or more commonly called the particle filter. Multiple particles are sampled from an initial distribution for the state space, and are assigned weights based on an importance function derived from the observation at that time step. Then, a resampling is taken from those particles with replacement and those particles are transitioned forward to the next time step using the state transition variables.

This filtering method does not make the assumptions of normality and linearity present in Kalman filters, but rather uses random samples (particles) from a probability density function. Additionally, it presents a remedy for particle degeneracy experienced in SIS. At a given iteration, all zero-weighted particles are eliminated from contention and N samples are taken from the remaining particles (with replacement). The importance weight is recalculated from the new observation. The result is N non-zero weighted particles remaining after each iteration.

Particle filtering does not require the explicit density function of the state space to be defined as time progresses, yet this methodology provides an "algorithm for propagating and updating these samples" (Gordon, Salmond, and Smith 1993). In fact, there are only three requirements to construct a particle filter. As before, let x_t be the state vector at iteration t, y_t is the observation vector at iteration t, and ε_t is the noise vector (with mean zero) associated with x_t. The requirements for particle filtering from Gordon et al. (1993) are as follows:

(a) A sample can be taken from the initial PDF at time $t = 1$ expressed as $p(x_1)$.
(b) The conditional PDF of the observation given the state at time t has a known form expressed as $p(y_t|x_t)$.
(c) A sample can be taken from the PDF of the state error term represented as $p(\varepsilon_t)$.

Like Kalman filtering, particle filtering is an estimation problem of the state variable vector from an observation vector. The state variable transition function is given by the following equation:

$$x_t = f(x_{t-1}, \varepsilon_{t-1}), \tag{3.7}$$

where f is the transition function. Unlike the state equation of the Kalman filter there is

19

no requirement that limits f to a linear function. Additionally, there is no assumption of normality in ε_t. Analogously, the observation equation is written using a function h:

$$y_t = h(x_t, \gamma_t). \tag{3.8}$$

Again there is no assumption that h is a linear measurement function, or that the measurement error γ_t is normally distributed. The mathematical intuition for the requirements (a) through (c) is revealed in the following statements summarized from Gordon et al. (1993) and Doucet et al. (2001). From requirement (a), there is a set of N random samples from our PDF at iteration t: $p(x_t)$. The conditional distribution of x_t given the observations up to time t using requirement (b) and following Bayes theorem is represented as:

$$p(x_t|y_{0:t}) = \frac{p(y_t|x_t)p(x_t|y_{0:t-1})}{\int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t}, \tag{3.9}$$

where $p(y_t|x_t)$ represents the probability density of the observation y_t given the state x_t, which is not explicitly known. Let $\tilde{x}_t$ be a random variable generated from $p(x_t|y_{0:t})$ and let $\tilde{x}_t^{(i)}$ refer to an individual sample i. Similarly with requirement (c), $\tilde{\varepsilon}_t$ is sampled from its distribution $p(\varepsilon_t)$. The state equation is used to get the prediction of the next state:

$$\tilde{x}_{t+1} = f(\tilde{x}_t, \tilde{\varepsilon}_t). \tag{3.10}$$

Requirement (b) is used to calculate the weight assigned to each particle i, expressed as:

$$w_t^{(i)} = \frac{p(y_t|\tilde{x}_t^{(i)})}{\sum_{j=1}^{N} p(y_t|\tilde{x}_t^{(j)})}, \tag{3.11}$$

where $w_t^{(i)}$ is the normalized weight of each particle. This weighting scheme is used for resampling N particles to be used at iteration $t+1$. The next set of particles $x_{t+1} \sim p(x_{t+1}|y_{0:t})$, is derived using the recursive function (3.10).

Particle filtering has several benefits. The first is that the algorithm is easy to implement while staying computationally simple. It can be implemented in a variety of programming languages. Additionally, it offers highly flexible and adaptable options due to its nonparametric nature, and because any weighting function can be used.

3.2.2 Basic Particle Filter Algorithm

A PF was introduced as a Monte Carlo method for predicting the state variable at a future state from a collection of observations at previous iterations. In this section, the notation for a basic particle filter algorithm is formalized. In this algorithm, $\tilde{x}_t^{(i)}$ is the sampled particle i at iteration t, $w_t^{(i)}$ is the weight of particle i at iteration t, y_t is the observation at iteration t, and N is the number of particles. At $t = 1$ the particles are sampled from the probability distribution $q(x_1|y_1)$, which is the importance density function conditioned only on the first observation. Weights are then computed for each sample i using the distribution $p\left(\tilde{x}_1^{(i)}\right)$, observation density function $h\left(y_1|\tilde{x}_1^{(i)}\right)$, and the importance density function $q\left(\tilde{x}_1^{(i)}|y_1\right)$. Once a weight has been assigned to each particle, a resampling with replacement occurs where $x_t^{(i)}$ is the resampled particle that is brought forward to the next iteration and the weights are reset to be $1/N$. At all time iterations after $t = 1$, particles are sampled through the importance density function defined as $q(x_t|y_t, x_{t-1}^{(i)})$, now conditioned on prior states as well as the current observation. At these iterations, weights are now a function of the conditional state density function $f\left(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}\right)$ rather than the initial distribution. Again, the weighted particles are resampled with replacement and the resulting set of particles is brought forward to the next iteration with corresponding weights reset to be $1/N$. The notation $\left\{\frac{1}{N}, x_{1:t}^{(i)}\right\}$ denotes the set of pairs of weights and particles as in Doucet and Johansen (2009). A simple algorithm for PF is expressed as:

If $t = 1$:

- Sample $\tilde{x}_1^{(i)} \sim q(x_1|y_1)$
- Compute weights $w_1^{(i)} = \dfrac{p\left(\tilde{x}_1^{(i)}\right)h\left(y_1|\tilde{x}_1^{(i)}\right)}{q\left(\tilde{x}_1^{(i)}|y_1\right)}$
- Take N samples with replacement of $\left\{w_1^{(i)}, \tilde{x}_1^{(i)}\right\}$ to obtain $\left\{\frac{1}{N}, x_1^{(i)}\right\}$

If $t > 1$:

- Sample $\tilde{x}_t^{(i)} \sim q(x_t|y_t, x_{t-1}^{(i)})$
- Set $\left(x_{1:t-1}^{(i)}, \tilde{x}_t^{(i)}\right) \longrightarrow \tilde{x}_{1:t}^{(i)}$
- Compute weights $w_t^{(i)} = \dfrac{h\left(y_t|\tilde{x}_t^{(i)}\right)f\left(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}\right)}{q\left(\tilde{x}_t^{(i)}|y_t, \tilde{x}_{t-1}^{(i)}\right)}$
- Take N samples with replacement of $\left\{w_t^{(i)}, \tilde{x}_{1:t}^{(i)}\right\}$ to obtain $\left\{\frac{1}{N}, x_{1:t}^{(i)}\right\}$.

3.3 Issues that Arise with Particle Filters

The simplicity, flexibility, and speed of a PF come at a cost of particle degeneracy and the curse of dimensionality. Particle degeneracy is the propensity of a set of particles in a PF to reduce to one non-zero weighted particle. In addition, a PF suffers from the curse of dimensionality. The curse of dimensionality is described as an exponential increase in computational complexity as dimensionality increases. This issue is present in all nonlinear filters. This section discusses degeneracy, the curse of dimensionality, and possible solutions to these issues.

3.3.1 Degeneracy

An issue that affects all particle filters is particle degeneracy. Particle degeneracy remains a prevalent issue because the resampling solution posed by Gordon et al. (1993) and discussed in Section 3.2.1 only masks the problem. In that paper, a remedy to the classical particle degeneracy in SIS is presented, where instead of sampling directly from the appropriate empirical distribution, N samples are taken with replacement from the remaining non-zero weighted particles at each iteration. With this methodology, a zero-weighted particle is never brought to the next iteration. Unfortunately, the issue with particle degeneracy is not solved but rather just masked. While it is true that there are N non-zero weighted particles iterated forward, one cannot assume that there are N unique particles.

When resampling N elements with replacement, in order to probabilistically achieve the most unique sample set of size N, the probability weightings of each particle should be equal. If the probability of selecting one element was higher than the rest, then it would have a higher probability of appearing in the sampled set more than once, thus diminishing set uniqueness. Now assume a set S with n distinct elements. Using Stirling numbers of the Second Kind, $S_2(n, m)$, the probability of having sampled m unique elements from the set S (given a sampling of n times with replacement) is as follows (Henry 2011, Weisstein 2016):

$$P[m] = \frac{S_2(n, m)n!}{n^n(n - m)!},$$ (3.12)

where $S_2(n, m) = \frac{1}{m!} \sum_{i=0}^{m}(-1)^i \binom{m}{i}(m - i)^n$. The expected number of unique elements is:

$$E[m] = n\left(1 - \left(1 - \frac{1}{n}\right)^n\right),$$ (3.13)

with a variance:

$$Var[m] = n\left(1 - \frac{1}{n}\right)^n + n^2\left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right) - n^2\left(1 - \frac{1}{n}\right)^{2n}. \tag{3.14}$$

Observing uniqueness as a proportion of the original set yields the following result: $E\left[\frac{m}{n}\right] = \left(1 - \left(1 - \frac{1}{n}\right)^n\right) \approx 0.63$ as $n \to \infty$. Therefore, uniqueness of the set degrades by approximately 37% at each iteration. This is actually a lower bound on the degeneracy because in the next iteration there may be fewer than n unique elements. Although each element of the set has equal probability, each unique element has a different probability as a linear function of how many repetitions appear in the set. For example, say x_i has 4 repetitions and x_j is unique, therefore x_j has a probability of $1/n$ while x_i has a de facto probability of $4/n$. This difference in probability exacerbates the degeneracy in the next iteration. This issue is always prevalent, and there is no direct method of resolving it.

3.3.2 Curse of Dimensionality

Like all nonlinear filters, a PF suffers from the curse of dimensionality, where the computational complexity of the filter is expected to increase exponentially with the dimensionality of the problem. The dimensionality of the filter is directly related to the dimensionality of the state space. For example, the state space may be position, time, and temperature, implying a three dimensional problem. Unfortunately, there is no closed form solution for this increase in computing. However, the increase is expected to be comparable to Monte Carlo integration techniques. The number of particles, or N, is required to be exponentially higher when operating at increased dimensions.

There are many types of particle filters, and the one introduced by Gordon et al. (1993) is considered a basic algorithm, which suffers tremendously from the curse of dimensionality. Some of the advanced methods introduced by Doucet and Johansen (2009) have significantly smaller increases in computational complexity thus making them appropriate for higher dimensions. Daum and Huang (2003) run experiments between two filters, basic and advanced, where an advanced filter will have the embellishments described later in this chapter. They test the run times of each filter while increasing the dimensionality. For example, they show that the advanced filter with 20 dimensions has a computational run time of approximately one second, whereas the basic filter with six dimensions has a

computational run time of over 30 seconds.

3.3.3   Dual Relationship between Degeneracy and Dimensionality

The experiments by Daum and Huang (2003) reveal an insight into the relationship between the curse of dimensionality and particle degeneracy. The embellishments in the advanced filter are implemented to combat particle degeneracy, yet there is a substantial decrease in the effects of dimensionality. As the dimensionality increases, the number of particles required for the convergence in the distribution of the particles to the true distribution of the state space increases. Yet if no safeguards against degeneracy are implemented, an additional increase in unique particles is required. This results in some negative feedback between the two issues that can be exploited. This dual relationship allows us to mitigate the curse of dimensionality by focusing on reducing degeneracy.

3.4   Remedies for Degeneracy

Brute force methods, like increasing the number of particles, may not be computationally feasible to be a viable solution for particle degeneracy. Dimensionality is often specified by the problem and is difficult to overcome, but particle degeneracy can be sometimes be addressed with relative ease. Fortunately, the dual relationship between particle degeneracy and the curse of dimensionality implies that a mitigation technique applied to one can possibly mitigate the other. Although there are no direct solutions to the issue of degeneracy in a particle filter, there have been a few mitigation techniques offered, and they are described in this section.

3.4.1   Roughening

The first technique to mitigate degeneracy offered by Gordon et al. (1993) is called roughening or jittering. Specifically, roughening involves adding a slight Gaussian noise to each particle after its resampling procedure, thus distinguishing multiple replications of the same particle. This jitter is often normally distributed with a mean of zero and a standard deviation given by the following equation:

$$\sigma_{jitter} = KEN^{1/d}, \tag{3.15}$$

where E is the distance between the minimum and maximum particle, d is the dimension of the PF, N is number of particles and K is a tuning parameter that the user selects. As K increases, so does the "spread" of the jitter. However, there is no requirement for the jitter to be normally distributed. An intuitive roughening technique often emulates the PDF of the distribution.

3.4.2   Prior Editing

Another technique by Gordon et al. is called prior editing. In this method, a predetermined acceptance algorithm is created, most likely from the importance density function. This technique increases the number of unique particles by running each particle through the acceptance algorithm. Each particle is sampled and roughened. The particles are each run through the acceptance algorithm to estimate its acceptance likelihood, i.e., its propensity to be resampled. If the particle has a low likelihood of acceptance, it is rejected and discarded. This process continues until N particles are accepted. A side benefit of this methodology is that information can be derived based on the percentage of particles rejected. However, in areas where the sampling distribution is spread out, the number of rejected particles will increase exponentially due to the low acceptance probability of each particle.

3.4.3   Adaptive Resampling

Adaptive resampling is a technique offered by Doucet and Johansen (2009). The intuition behind this technique is to only conduct resampling of particles at certain times, thus reducing the degeneracy as well as the computational complexity. There is a specific criterion that must be achieved to trigger a resampling. If the criterion is not met, then all particles are carried to the next iteration. If the criterion is met, the PF progresses as normal with the combined weights from previous iterations. The main benefit to this methodology is that when the system is less variable (like driving on a straight road in the GPS example) there are fewer opportunities to trigger resampling resulting in an extremely fast algorithm. However, if the system is highly variable and requires many iterations of sampling, the benefits of adaptive resampling are minimal.

There are no limits on the number of particle degeneration mitigation techniques that one can implement. Due to strengths and weaknesses present in each technique, employing a variety of methods makes a PF algorithm more robust. Care should be taken, however,

to ensure that these techniques do not interfere with one another, i.e., if the roughening procedure consistently roughens a particle so that it fails the acceptance test in prior editing.

3.5   Algorithm

An enhanced particle filter algorithm that employs the previously describe techniques for mitigating particle degeneracy is discussed in this section. This algorithm builds on the one created by Gordon et al. (1993) described in Section 3.2.2. It incorporates roughening procedures as well as adaptive resampling techniques. Using the standard established by Daum and Huang (2003), this algorithm is considered an advanced algorithm with features that mitigate particle degeneracy and thus reduce the curse of dimensionality.

At $t = 1$, the particles are sampled from the probability distribution $q(x_1|y_1)$, which is the importance density function conditioned only on the current observation. Weights are then computed for each sample i using the distribution $p\left(\tilde{x}_1^{(i)}\right)$, observation density function $h\left(y_1|\tilde{x}_1^{(i)}\right)$, and the importance density function $q\left(\tilde{x}_1^{(i)}|y_1\right)$. At this point adaptive resampling may be triggered. If triggered, a resampling with replacement occurs to generate new values of $x_t^{(i)}$. The particle is then applied to a roughening procedure defined as $R(x_t^{(i)})$. This roughening yields a particle defined as $x_t^{*(i)}$ that is iterated forward with weight $1/N$. If adaptive resampling is not triggered, the weights equalize across particles and the algorithm moves to the next iteration.

At $t > 1$, particles are sampled with the importance density function defined as $q(x_t|y_t, x_{t-1}^{(i)})$, now conditioned on the prior state as well as the current observation. This sample is added to the particle space. At these iterations, weights are now a function of the state conditional density function $f\left(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}\right)$ rather than the initial distribution. Again, if the adaptive resampling criteria is triggered, the weighted particles are resampled with replacement and roughened. The newly roughened particle is iterated forward and the weights are equalized. If adaptive resampling is not triggered, the weights are equalized and the algorithm moves forward. Here is the algorithm:

If $t = 1$:

- Sample $\tilde{x}_1^{(i)} \sim q(x_1|y_1)$

- Compute weights $w_1^{(i)} = \dfrac{p\left(\tilde{x}_1^{(i)}\right)h\left(y_1|\tilde{x}_1^{(i)}\right)}{q\left(\tilde{x}_1^{(i)}|y_1\right)}$
- If adaptive resampling criteria triggers resampling:
  - Take N samples with replacement of $\left\{w_1^{(i)}, \tilde{x}_1^{(i)}\right\}$ which yields $\left\{\frac{1}{N}, x_1^{(i)}\right\}$
  - Roughening: $x_1^{*(i)} \sim R(x_1^{(i)})$.
  - Set $\tilde{x}_1^{(i)} \leftarrow x_1^{*(i)}$
  - Set $\left\{w_1^{(i)}, \tilde{x}_1^{(i)}\right\} \leftarrow \left\{\frac{1}{N}, x_1^{(i)}\right\}$
- Else:
  - Set $\left\{w_1^{(i)}, \tilde{x}_1^{(i)}\right\} \leftarrow \left\{\frac{1}{N}, x_1^{(i)}\right\}$.

If $t > 1$:

- Sample $\tilde{x}_t^{(i)} \sim q(x_t|y_t, x_{t-1}^{(i)})$
- Set $\left(x_{1:t-1}^{(i)}, \tilde{x}_t^{(i)}\right) \longrightarrow \tilde{x}_{1:t}^{(i)}$
- Compute weights $w_t^{(i)} = \dfrac{h\left(y_t|\tilde{x}_t^{(i)}\right)f\left(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}\right)}{q\left(\tilde{x}_t^{(i)}|y_t, \tilde{x}_{t-1}^{(i)}\right)}$
- If adaptive resampling criteria triggers resampling:
  - Take N samples with replacement of $\left\{w_t^{(i)}, \tilde{x}_t^{(i)}\right\}$ which yields $\left\{\frac{1}{N}, x_t^{(i)}\right\}$
  - Roughening: Sample $x_t^{*(i)} \sim R(x_t^{(i)})$.
  - Set $\tilde{x}_t^{(i)} \leftarrow x_t^{*(i)}$
  - Set $\left\{w_t^{(i)}, \tilde{x}_{1:t}^{(i)}\right\} \leftarrow \left\{\frac{1}{N}, x_{1:t}^{(i)}\right\}$
- Else:
  - Set $\left\{w_t^{(i)}, \tilde{x}_{1:t}^{(i)}\right\} \leftarrow \left\{\frac{1}{N}, x_{1:t}^{(i)}\right\}$

In this algorithm there are no constraints placed on the importance function (which determines the weights), the adaptive sampling criteria trigger, or the roughening function. Thus the versatility of the PF is maintained while making it robust to dimensionality and particle degeneracy. This is the particle filtering algorithm that is implemented with the simulated Brownian bridge model. The specific weighting criteria, adaptive resampling criteria, and roughening procedures are described in Chapter 4.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Implementation

The purpose of this thesis is to estimate the distribution of target location between a start and end location based on intelligence updates along the projected path of travel. A simulated Brownian bridge model serves as the underlying stochastic model that generates a path that represents target motion. The particle filter is used to incorporate intelligence updates, which update the SBBM. This section addresses the process undertaken to implement a particle filter in conjunction with the SBBM.

4.1 Model Concept and Assumptions

Ultimately, the model employed in this thesis consists of two major components. The first component is a stochastic model to simulate the distribution of the target location inside an area of responsibility (AOR) over time. This thesis specifically looks at narcotics trafficking in the USSOUTHCOM AOR. There is a significant level of uncertainty in this context that is modeled with a few assumptions. The first assumption is that a narcotics trafficker (the target) will attempt to deliver his payload as efficiently as possible while attempting to remain covert. Additionally, while the exact departure and destination locations are unknown, they can be estimated to be within a specified area. Furthermore, the target's time frame for travel can also be estimated to be within a specified interval.

The second major component is determining an effective way to quantify intelligence updates. This includes both "negative" and "positive" updates, where a negative update is when the target is not observed in an intelligence area and a positive update implies the target was observed. The update type is modeled as a Boolean detection flag in the model. The Boolean detection flag is augmented with the probability of detection of the sensor region to quantify the effectiveness of the intelligence update. A negative update with high probability of detection may be preferable to a positive update with low probability of detection.

Additionally, there are some implicit assumptions made in the model. In operations, a positive confirmation of a target typically leads to interdiction by an asset that is able to

conduct that mission, i.e., a helicopter with authority for Airborne Use of Force. This model has the potential to provide decision makers with the ability to more effectively place assets in a position to conduct interdiction operations. However, this thesis makes the assumption that the search asset does not have the capability to interdict. If there was an asset to prosecute the target, the scenario/mission is complete no longer necessitating further use of the model. The model continues running after a potential interdiction to complete the heat map time series trajectory. Additionally, there are many Intelligence, Surveillance, and Reconnaissance assets that cannot interdict, i.e., a Maritime Patrol and Reconnaissance Aircraft (MPRA). This also accounts for sensors that can detect a target without visual confirmation, as is the case with electronic intelligence.

4.2   MATLAB Model

This thesis develops an extension to a time step stochastic model that was designed in MATLAB by Professors Dashi Singham and Michael Atkinson for the Center for Multi-Intelligence Studies (CMIS). The original SBBM is defined as the basic model. The model simulates a user specified number of Brownian bridges from a sampled start location to a sampled end location, where the location uncertainty areas determine the region of sampling. The basic model also incorporates a rudimentary intelligence updating algorithm that eliminates Brownian bridge paths that fail to meet a simple sensor criterion. Paths that do not enter the sensor area are eliminated in a positive update, and paths that enter the sensor area are eliminated in a negative update. The model creates a probabilistic heat map based on the concentration of paths meeting sensor intelligence, and then generates a series of plots showing the progression of heat maps at user-defined time steps. The model has several variants that include multiple signals or targets, multiple paths a target can take, as well as multiple intelligence sources. The primary focus for this thesis is the multiple intelligence sources variant. The enhanced model that uses a particle filter is henceforth called the advanced model.

4.2.1   User Inputs

The user inputs are broken down into three categories: model parameters, target data, and sensor data. The model parameters typically determine the fidelity of the model and, consequentially, the run time. The number of time steps, as well as the number

30

of simulated Brownian bridges, are model parameters. The computational run time is dependent polynomially on the number of Brownian bridges and the number of time steps in the simulated discretized Brownian bridge. The user also specifies the number of time snapshots that the model will generate, which corresponds to how many output plots are produced upon model completion. Also, the user is able to specify the variance of the Brownian bridges, which determines how spread out the simulated paths can become. A discussion on a method for calibrating the variance parameter is found in Cheng (2016).

For target data, the user defines both the start and end locations of the target's path with a level of location uncertainty. The speed of the target is a user-defined deterministic parameter. Deterministic speed is implemented to simplify the way time progresses in the model. However, any uncertainty regarding speed is incorporated into either location uncertainty or the Brownian bridge variance without much loss of fidelity. For example, if the speed of the target is highly uncertain (a target vessel could be a high-speed cigarette boat or low-speed Self Propelled Semi-Submersible), the user simply increases the uncertainty in the start and end locations, and increases the Brownian bridge variance parameter.

Sensor inputs are markedly different in the basic model and advanced model. The inputs are hard coded into the basic model, yet extracted from a comma separated value file in the advanced model. Only the sensor location, Boolean detection flag (positive or negative), and active sensor times are required for the basic model. An individual sensor area of coverage is represented by a rectangular area. This represents a patrol sector for a warship and/or maritime patrol plane. Additionally, the user determines when the sensor is active, whether at a single time unit in the basic model or a range of times in the advanced model. The Boolean detection flag is set equal to one if a positive intelligence sighting occurs in the sensor region when the sensor is active. In the advanced model, there is a probability of detection associated with each sensor. The run time is expected to increase with the number and time of active sensors due to an adaptive resampling embellishment of the PF, which will be explained. Table 4.1 describes a breakdown of the user inputs for both the basic and advanced model.

Table 4.1. Comparison of Sensor Inputs for the Basic and Advanced Model.

	Basic Model	Advanced Model
Input Methodology	Hard Coded	Input File
Active Sensor Time	Single Time	Time Range
Boolean Detection Flag	Yes	Yes
Probability of Detection	Implicit at 100%	User Defined

4.2.2 Intelligence Updates and Model Output

The basic model executes a simple intelligence updating algorithm. At each sensor location, a Boolean flag determines whether there is a positive intelligence sighting or not, as described in Section 4.2.1. At the sensor's active time, a positive Boolean detection flag causes all Brownian bridge paths that fall outside the sensor location at that time step to be removed from future realizations of the heat map. Conversely, a negative detection results in all the paths inside the sensor location at that time step to be removed in a similar manner. Figure 4.1 shows a set of output plots produced by the basic model. Sensor regions are defined by red boxes (which turn green during active time in the advanced model) for a given snapshot time found in the plot title. The scale of the right side corresponds to the range of probabilities (throughout the entire system) associated with the heat map. The moving mass represents the heat map that corresponds to the target location.
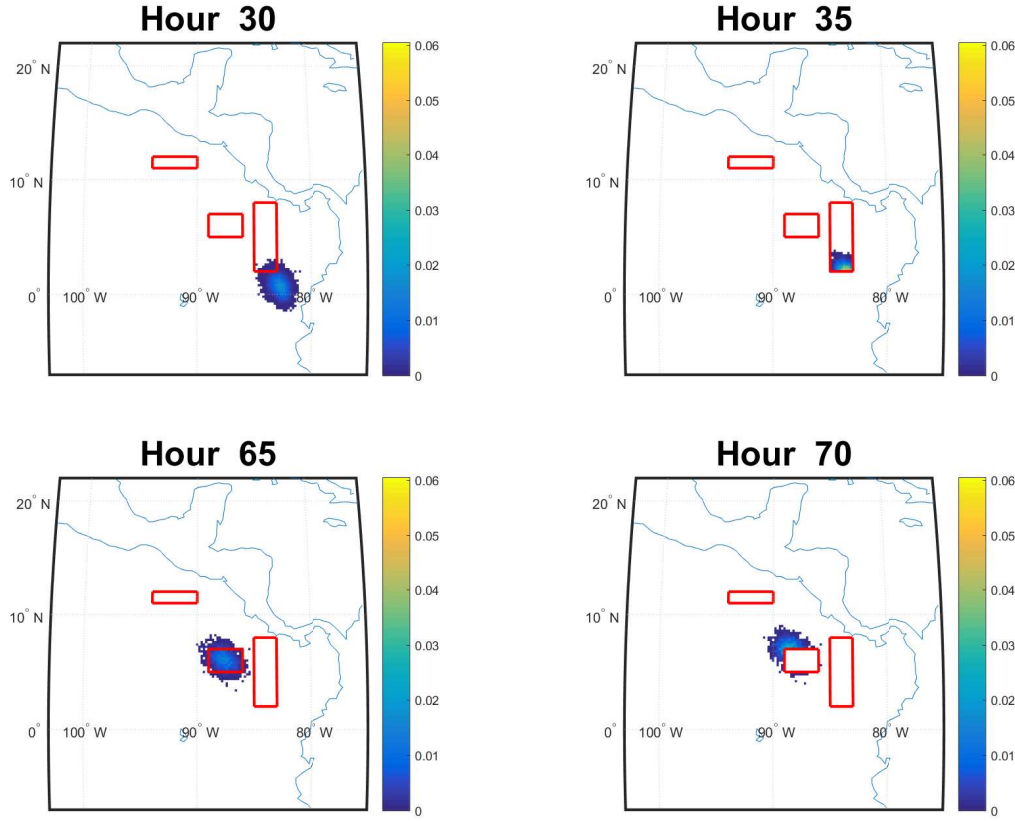
Figure 4.1. Intelligence Updates with the Basic Model ($\sigma^2 = 0.1$).

In the scenario in Figure 4.1, a positive detection is set for the lower right sensor at hour 35 and a negative detection is set for the central sensor at hour 70 to demonstrate the elimination of paths outside and inside of the sensors, respectively.

There are several issues with this methodology, the foremost being a lack of accounting for any probability of detection. These sensors are taken to be 100% accurate within their respective active times throughout their entire region allowing for no possibility of false negative or false positive detections. Even the additional embellishment by Cheng (2016) only incorporates a probability of detection via a Bernoulli random variable simulating positive or negative detections across many replications of the experiment. The second major problem with the updating method used in the basic model is the level of degeneracy experienced. This model creates 20,000 Brownian bridges, not to increase resolution, but to combat the degeneracy experienced via the elimination protocol. Daum and Huang (2003)

demonstrate that this level is excessive for a two-dimensional problem.

4.3 Incorporation of a Particle Filter

The method for incorporating intelligence updates in the basic model needs improvement. However, many components of the model are ideal for use with a particle filter. The SBBM generates a series of Brownian bridge paths that aggregate to estimate the initial PDF, and each path serves as a particle. Additionally, the probabilistic heat map provides a mechanism for visual representation of the distribution of particles.

From the basic model, the procedure for creating Brownian bridges is retained. Additional levels of fidelity are added by allowing the sensor to be active for a period of time rather than just at a specific time step. A probability of detection parameter is included for the active sensor time interval and works in conjunction with the Boolean detection flag for whether a target is observed. The active sensor time acts as a surrogate for adaptive resampling. If no sensor is active, the particle filter does not conduct a resampling and the simulated paths remain the same. When there is an active sensor, the particle filter conducts a resampling based on a weighting function that is calculated separately. Using this method of adaptive resampling, there is a significant reduction in the number of resamplings that occur. For example, assume the model calculates a total time in system of 150 model hours. Suppose there are three sensors and each operates for six hours. Then the sensors are active for 18 hours out of the 150-hour model, thus resampling only occurs at 12% time steps. This translates to an 88% reduction in number of resamplings, as opposed to resampling every time step. For a 500 time step model, this translates to only 60 resamplings conducted, a significant savings in run time.

4.3.1 Weighting Scheme

Ultimately, there is a need to establish two specific weighting schemes for the particle filter to determine the weight associated with each particle during a resampling step. One scheme corresponds to a negative detection and the other to a positive detection. Additionally, the probability of detection needs to be incorporated appropriately for that weighting scheme. The weighting scheme has the following inputs: the probability of detection, p_d, the Boolean detection flag D, and the index for the specific particle, i.

The baseline weight is set to $w_i = 1$ for all i, at the beginning of the experiment, and at all times when the weights across particles are equalized. A reasonable start point for the weight assignment is to use Euclidean distance. Suppose that the strongest relationship between the sensor and weight occurs in the center of the sensor range. Then weights are formulated using the Euclidean distance, $dist_i$, from the location of particle i to the Euclidean center of the rectangular sensor range. Consider the line ray starting at the sensor center and passing through particle i. Take $maxdist_i$ to be the length of the line segment created on this ray between the sensor center and point that crosses the sensor boundary, as shown in Figure 4.2. The normalized Euclidean distances are defined as $rd_i = \frac{dist_i}{maxdist_i}$. Letting rd_i be the foundation of the weight for particle i puts less weight on those particles inside of the sensor range since $rd_i \leq 1$ when $dist_i \leq maxdist_i$, so it is useful for weighting a negative detection. The inverse of rd_i is greater than one when the target is inside the sensor range, so the inverse can be used to assign weights in a positive detection.
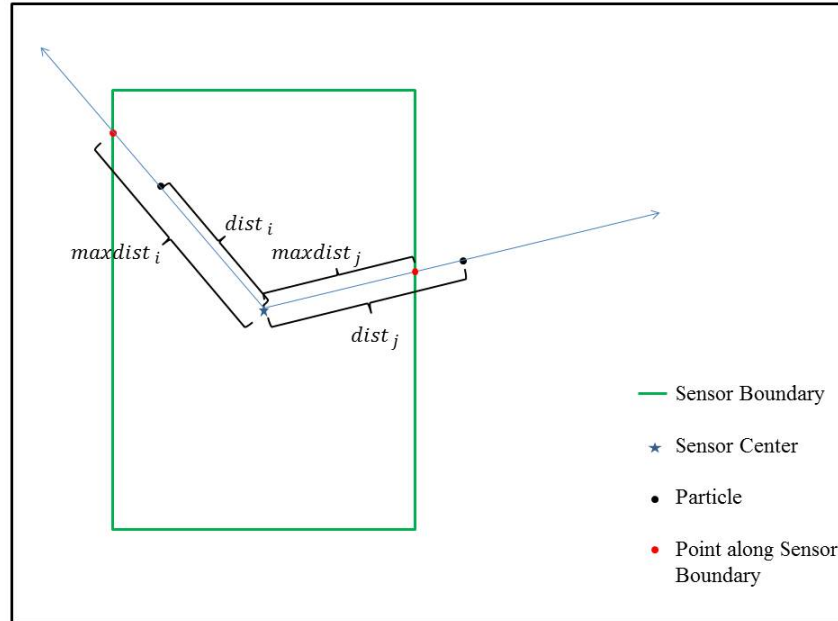

Figure 4.2. Visualization of Terms for Weighting Scheme.

With an established starting point for the weighting scheme, the next step is to incorporate the probability of detection of the sensor. An ideal weight function that incorporates the probability of detection is one that approaches our baseline weight when the probability of

detection approaches zero, yet corresponds to a proportional weighting scheme (inside or outside of the sensor) as the probability of detection approaches one. Raising the nominal weighting scheme by the probability of detection does precisely that, specifically $(rd_i)^{p_d}$ when there is a negative detection and $\left(\frac{1}{rd_i}\right)^{p_d}$ when there is a positive detection. Lastly, these two scenarios are combined into a single equation by using the Boolean detection flag, which defines $D = 1$ for a positive update and $D = 0$ for a negative update, as such:

$$w_i = \left((1 - D)rd_i + \frac{D}{rd_i}\right)^{p_d}.$$ (4.1)

4.3.2   Roughening Procedures

The adaptive resampling measure corresponding to active sensor times does not mitigate particle degeneracy enough to be effective. Testing shows degeneracy of 97-98% of our starting particles by the end of a run. Obviously, additional degeneracy mitigation techniques are necessary. From Section 3.4, there are two options available: prior editing and roughening. Prior editing is not feasible since it would require the creation of an entire Brownian bridge for each run through its acceptance test and the computational run time is predicated on the number of Brownian bridges created. Thus, the only realistic option remaining is the roughening technique.

Gordon, Salmond, and Smith (1993) suggest adding a slight Gaussian jitter based on a number of factors described in Section 3.4. However, there is little justification for this methodology other than just to jitter the observation. Due to the time series and Markovian nature of the Brownian bridge, a more robust technique is obvious. For a given particle, a two-dimensional Brownian bridge was created from a start point and an end point. At any given time, only the particle's present position along the path has any impact on its future position. Thus at every iteration the remaining points of the Brownian bridge can be simulated as a new Brownian bridge with the present position as the "start" point. The property can be used for the robust roughening technique described next.

This method of roughening involves creating a Brownian bridge from the point the sample was taken using the current position of the particle as the start point and the end point of the particle as the new end point. The ingenuity behind this method is the ability to

jitter the new particle while maintaining the same distribution of the PDF. Figure 4.3 shows a one-dimensional roughening of a Brownian bridge when resampling occurred at time $t = 0.2$.
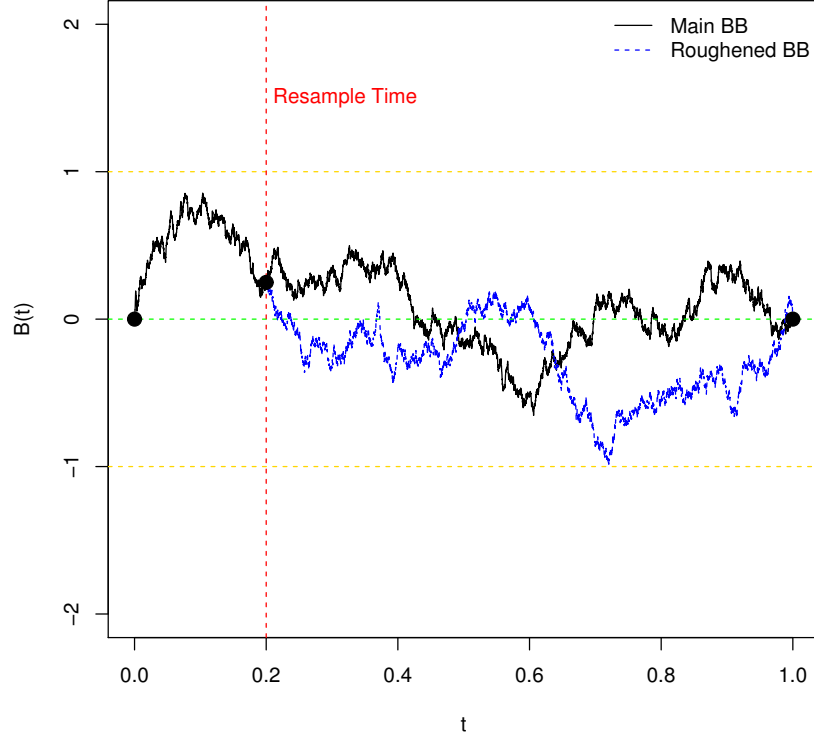

Figure 4.3. Example of a Simulated One-Dimensional Standard Brownian Bridge with a Particle Filter Roughening Procedure Applied.

In Figure 4.3, a split occurs at time $t = 0.2$, yet both Brownian bridges eventually arrive at the original endpoint. A similar method is executed in two dimensions for the roughening procedure of our model. The roughening procedure occurs at each point a resampling is conducted. After resampling, each resampled particle becomes the starting point for the next Brownian bridge maintaining its current endpoint. The original Brownian bridge combines with a new Brownian motion process to create a new Brownian bridge with the

following equations:

$$\tilde{B}_{i,x}(t) = B_{i,x}(\bar{t}) + \left(B_{i,x}(T) - B_{i,x}(\bar{t})\right)\frac{t - \bar{t}}{T - \bar{t}} + \tilde{W}_{i,x}(t) - \tilde{W}_{i,x}(T)\frac{t - \bar{t}}{T - \bar{t}}, \; t \in (\bar{t}, \, T),$$

$$\text{(4.2)}$$

$$\tilde{B}_{i,y}(t) = B_{i,y}(\bar{t}) + \left(B_{i,y}(T) - B_{i,y}(\bar{t})\right)\frac{t - \bar{t}}{T - \bar{t}} + \tilde{W}_{i,y}(t) - \tilde{W}_{i,y}(T)\frac{t - \bar{t}}{T - \bar{t}}, \; t \in (\bar{t}, \, T),$$

where $\bar{t}$ is the current time step, T is the final time, $\tilde{W}_{i,x}$ is the newly created Brownian motion process over time range $(\bar{t}, T)$ with the same parameters as the original Brownian bridge, and $B_{i,x}$ and $\tilde{B}_{i,x}$ are the original and roughened Brownian bridge, respectively, for the x-coordinates of particle i. The roughening is conducted similarly for the y-coordinates. With this roughening procedure, the expected 37% particle uniqueness loss at time t when the sampling occurred is regained by time $t + 1$ with the new Brownian bridges.

Implementing the changes described above and using similar parameters to the basic model, adjustments are made to the probability of detection and active times of each sensor. In the next example, the advanced model runs with each sensor probability of detection set to 0.60 and sensor active time set to 6 hours, and the results are shown in Figure 4.4.
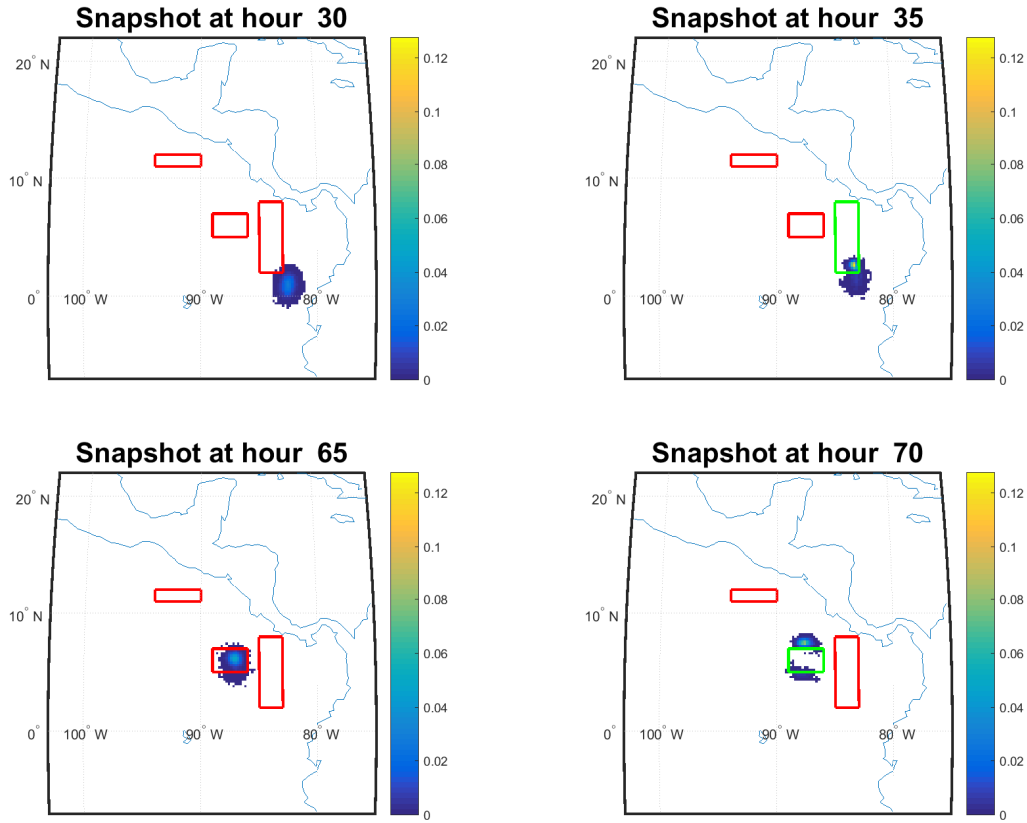
Figure 4.4. Advanced Particle Filter Model with Probability of Detection of 0.60 ($\sigma^2 = 0.1$).

Even with a slightly reduced probability of detection (from $p_d = 1$), there is already a visible change. The advanced model does not entirely eliminate particles based on whether they fell inside or outside of the sensor range. This development incorporates realism into the model as the particles that were not eliminated could represent false positives or false negatives, as appropriate. Further investigation shows how a significant reduction in the probability of detection affects the particles as seen in Figure 4.5.
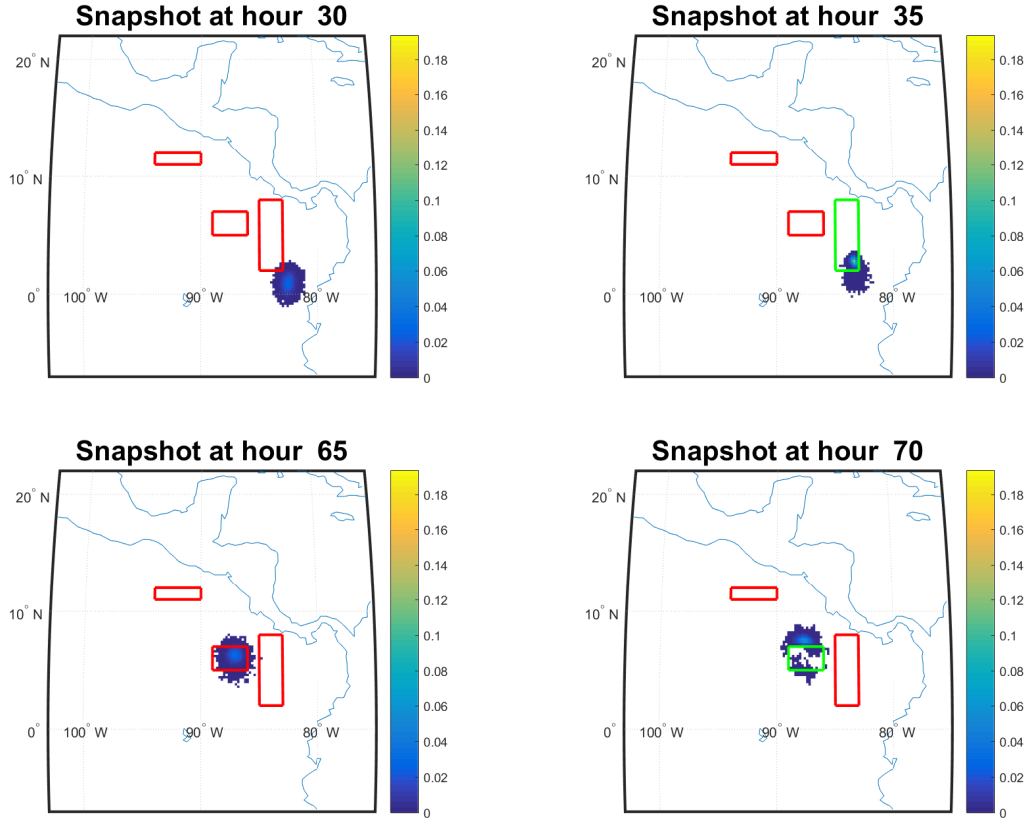
Figure 4.5. Advanced Particle Filter Model with Probability of Detection of 0.25 ($\sigma^2 = 0.1$).

With a probability of detection of 0.25, the particle filter only registers a small change in the heat map when sensor information is collected. This is in contrast to when the probability of detection approaches one, when there is a higher propensity to mirror the basic model. With a probability of detection near zero, the model would act similar to one with no sensors.

4.4   Experimentation

Using the advanced model with the particle filtering method, experiments were conducted to determine how well the model reacts to changes in the inputs with a more realistic scenario. Unless specified to the contrary, the number of Brownian bridges and time steps are set to 20,000 and 500, respectively. These levels allow for the best tradeoff between model fidelity and run time. Run times of one instance at these levels typically fell between 60-120

seconds depending on the number of sensors and corresponding active sensor times.

4.4.1 Intelligence adjustments

The toy examples presented so far have used three sensors to demonstrate the sampling methodologies of our two models, basic and advanced. However, there is a level of fantasy in the enormous size of the sensor area. In Figures 4.1, 4.4, and 4.5, the sensor areas from lower right to upper left are larger than 57000 $miles^2$, 28000 $miles^2$, and 18000 $miles^2$. In other words, the area sizes are comparable to Alabama, West Virginia and the combined size of Maryland and Delaware, respectively. These sensors are purposefully unrealistically large in order to highlight differences in the two models. Now, in order to better demonstrate the advanced model, a reduction in sensor size is needed.

To determine the best sensor size, the first step is to select an appropriate platform. In the counter narcotics mission, there are primarily three platforms used, maritime patrol aircraft, warship, and helicopter. Helicopters are dropped from contention due to their primary function as an interdiction platform given their on station time, range and dependence on warships. This platform is appropriate for a high probability of detection in a small area for a small duration. A warship provides the longest on station time of all the platforms, but its slow speed gives credence to a very low probability of detection while in a large area for a long duration. An MPRA provides the best of both worlds and adheres to the previously made assumption of non-interdiction. With this platform, you get a moderately large probability of detection for relatively large search areas. The document from the Office of the Chief of Naval Operations (1997), or, SAR TACAID, provides us with the most efficient regime of flight. Assume a target with size of 25 feet with visibility at 10 sm. The optimal altitude to maximize the visual range of a fixed wing aircraft is 3000 feet at 180 kts (207 mph). Visually, that allows for an effective range of approximately four miles and a line of sight range for radar of approximately 67 miles, as can be approximated using $RadarRange \approx 1.23\sqrt{Altitude}$. Suppose there is an on-station time of 6 hours. This allows for a visual search range of approximately 9900 $miles^2$, and an 80% probability of detection (Office of the Chief of Naval Operations 1997). This is used as the baseline for setting probability of detections and area sizes. Figure 4.6 provides a quick reference for determining the probability of detection from the coverage factor, which is defined as "the ratio of the search effort to the area searched" (National Search and Rescue Committee
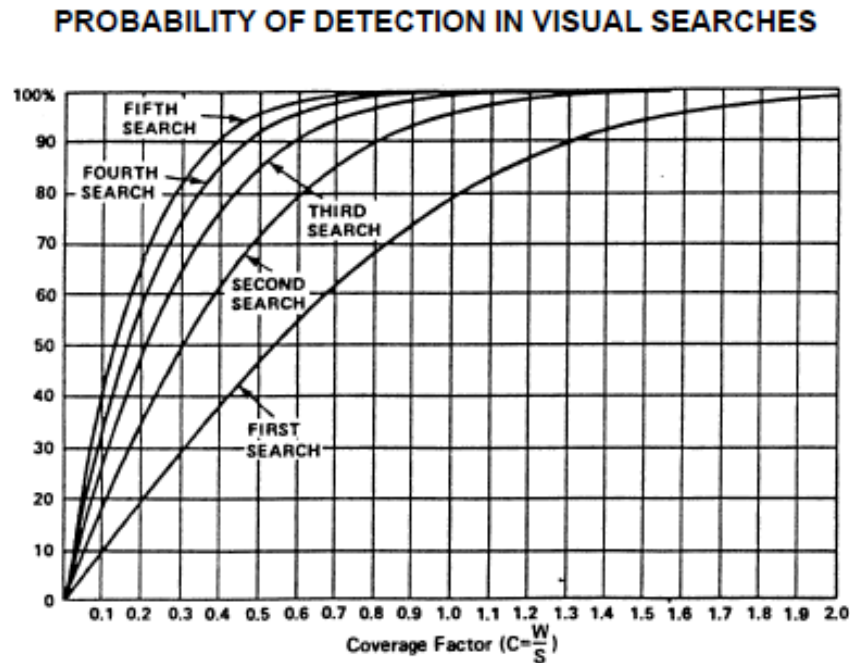
2000).

Figure 4.6. Probability of Detection vs. Coverage Factor from Navy SAR TACAID. Source: Office of the Chief of Naval Operations (1997).

An established baseline coverage area provides information on a more realistic patrol scenario. The following table provides a guide for probabilities of detections given a number of patrol assets and the size of the sensor area. Note that the sensor areas addressed in Table 4.2 are labeled from lower right to upper left.

Table 4.2. Probability of Detection of Baseline Sensor Areas.

Sensor	Size ($miles^2$)	Probability of Detection		
		One MPRA	Two MPRA	Three MPRA
Lower Right	57000	0.17	0.35	0.42
Center	28000	0.33	0.55	0.70
Upper Left	18000	0.50	0.75	0.87

A significant amount of resources must be used in order to achieve a reasonable probability of

detection. If this is not feasible, then reducing the sensor area also increases the probability of detection. A more realistic model is established when following this criterion. In Figure 4.7, the three patrol boxes are scaled to an appropriate size ($\sim$10000 $miles^2$) for an active time of six hours to facilitate a probability of detection of 0.80.
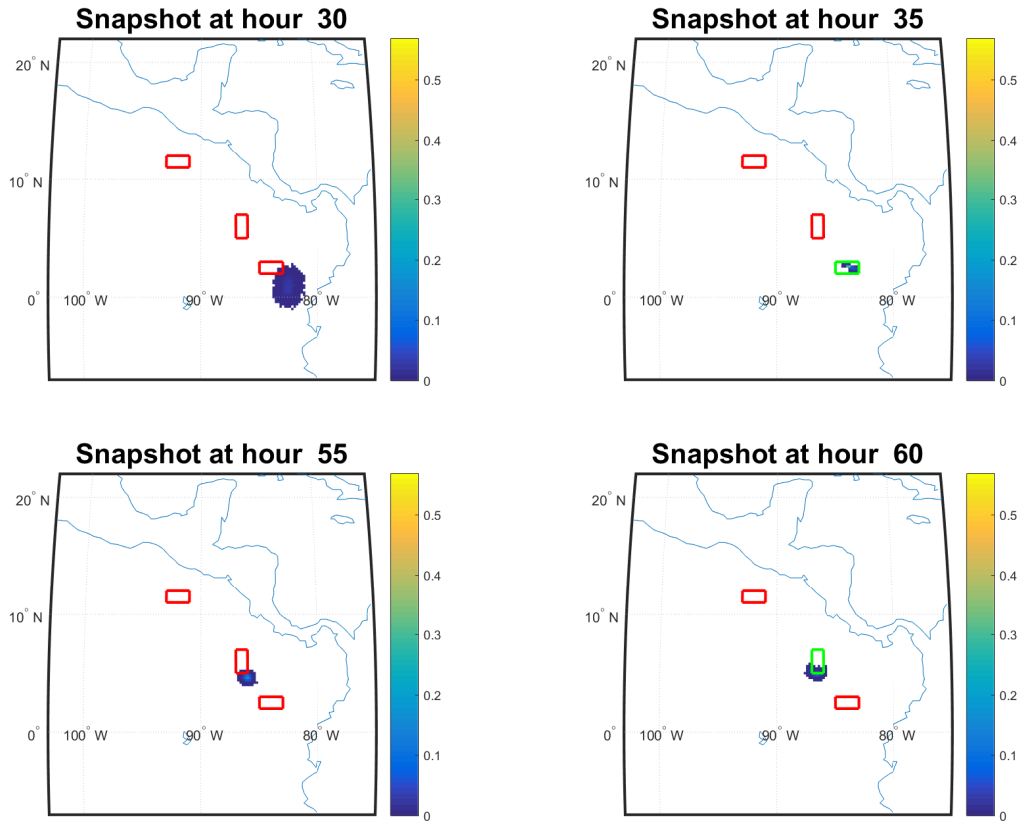

Figure 4.7. Advanced Model with Realistic Patrol Areas and Probability of Detection of 0.8 ($\sigma^2 = 0.1$).

4.4.2 Weighting Scheme Adjustments

The most significant way that an individual can change the structure of a particle filter algorithm is by changing its weighting scheme. Currently the model uses a weighting scheme described by (4.1). This weighting scheme, while appropriate for a lower probability of detection, tends to become unstable as the probability of detection approaches one, specifically in cases of positive intelligence. Since the weight is dependent on the inverse of

the Euclidean distance from the center, there exists a possibility of an explosion in the particle weight if that particle falls near the center. During testing, high weights of approximately 1500-2000 (relative to the original baseline weight of $w_i = 1$) are not uncommon. Since the baseline weight is set as one, this high weight results in oversampling near the sensor center. For positive intelligence, the goal is to focus the majority of the weight on particles inside the whole sensor area, not just near the center. There are two ways to approach this problem: the first method involves an adjustment to the weighting equation and the second involves a simultaneous change to the weighting equation and the adaptive sampling algorithm.

In order to foster seamless incorporation into the current model, the baseline weight remains the same ($w_i = 1$). With that assumption, the primary focus is on particle i such that $\frac{dist_i}{dist_{max}} = rd_i \leq 1$ since the change is applied to positive intelligence samples. The goal is to set the weights outside the sensor to have a uniform weight of $w_i = 1$, and those inside the sensor to have weights between one and two, with higher weights associated with proximity to the center of the sensor. To do this, the weighting for positive intelligence is modified to be $2 - \min\{1, rd_i\}$. This weight scheme for a positive update ensures that particles beyond sensor range maintain a weight of one and the particles inside the bounds have a weight between one and two. Substituting this equation for positive intelligence in (4.1) yields the following new weighting scheme:

$$w_i = ((1 - D)rd_i + D(2 - \min\{1, rd_i\}))^{p_d} . \tag{4.3}$$

With this new weighting scheme, the overweighting of particles that randomly fall close to the center point of the sensor range is mitigated. Figure 4.8 shows the model with the new weighting scheme.
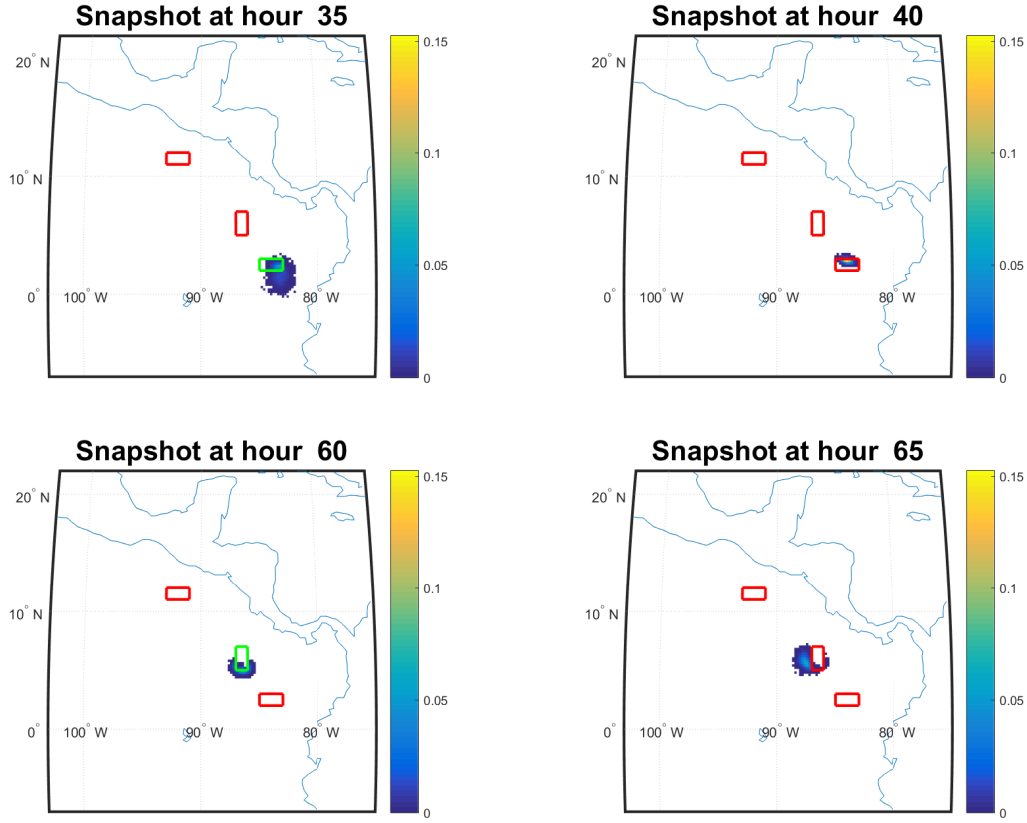
Figure 4.8. Advanced Realistic Model with Probability of Detection of 0.8 and Updated Weighting Scheme ($\sigma^2 = 0.1$).

A quick comparison between Figures 4.7 and 4.8 shows how the new weighting scheme affected the heat map. In the update, the probability mass is more spread out over a larger area at hour 35, representing greater uncertainty in where the target is located than when the weight is concentrated near the center.

Another option available to use in conjunction with the weighting scheme is to incorporate adaptive resampling. Unlike the adaptive resampling used thus far, this method allows for weights to be continually adjusted until a resample takes place, and only at that point will the weights reset to one. With the weight scheme defined by (4.3), no weight is larger than 2^{p_d} or smaller than one for a positive update, which does not translate to much difference in probability. If a more pronounced sampling effect is desired, delay resampling until a certain threshold is reached. For example, a threshold might be defined as a requisite

number of time steps during active sensor time before resampling. In this case, a particle's weight is accumulated at each iteration where no resampling occurs. Suppose a threshold of five time steps serves as a baseline time until resampling. Under this weighting scheme, the smallest cumulative weight possible at the fifth time step is five and the largest is $5 * 2^{p_d}$, which corresponds to a wider range of probabilities and thus a more pronounced sampling effect. Figure 4.9 represents the model with this threshold weighting scheme.
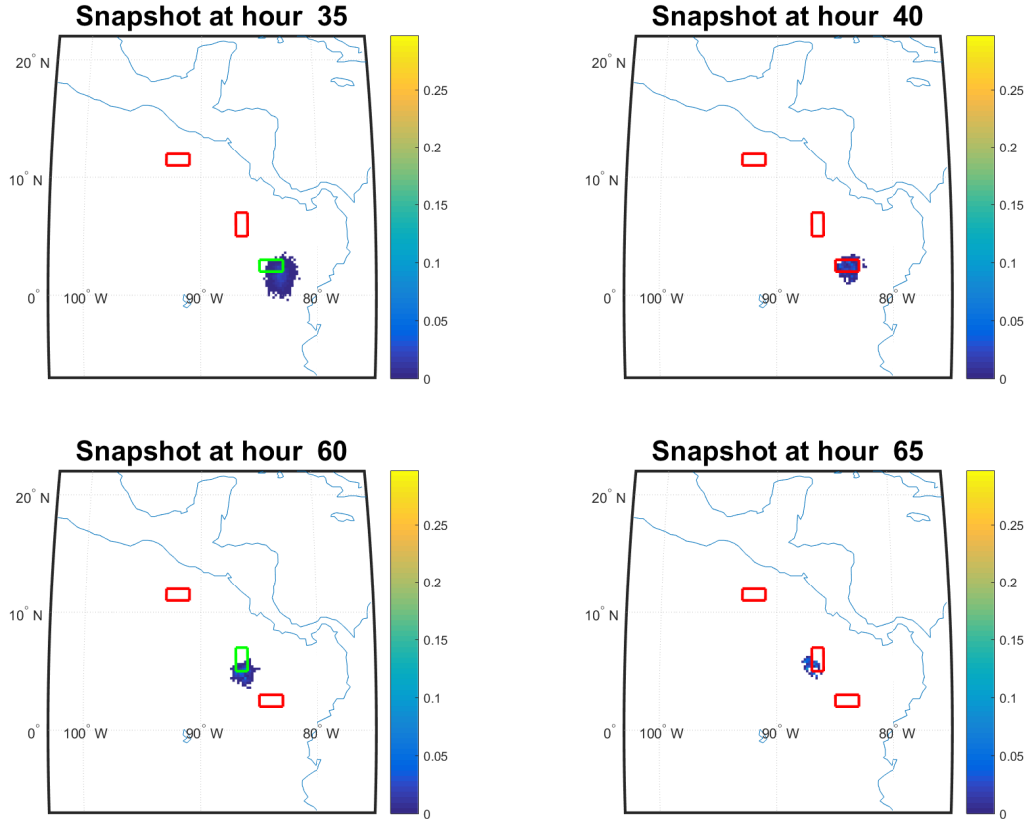

Figure 4.9. Advanced Realistic Model with Probability of Detection of 0.8 and Updated Threshold Weight Scheme ($\sigma^2 = 0.1$).

The adaptive sampling method incorporated up to this point has limited resampling to be conducted only when sensors are active. This has significantly reduced the number of resamplings conducted. The implementation of a threshold weighting scheme further reduces the number of resamplings. Given that patrol times account for 12% of the model run time in this example, the incorporation of threshold weighting (specifically a five step

threshold) further reduces the number of resamplings conducted by approximately 80%. This reduces the total number of resamplings conducted to 2% of all time steps. Therefore, in order to conduct an effective number of resamplings, the number of time steps for each simulated Brownian bridge must be increased from 500 to 2000 in this case. Additionally, the standard roughening technique will only be applied at the resampling iterations. A comparison of Figure 4.9 against Figure 4.8 shows there is a loss of smoothness due to a combination of increased time steps and the 80% reduction in resampling.

Although the two weighting techniques produced similar results, the weighting technique that does not implement a time step threshold is preferable. It produces smoother results that are visibly more tractable. It eliminates a possible source of bias in the model by not requiring the user to arbitrarily decide a structural parameter of the model, i.e., the threshold level. Most importantly, it does not require a trade-off in computational run time to achieve a reasonable level of fidelity as the number of time steps is one of the main factors that contributes to increased run times.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions and Recommendations

An effective way to incorporate intelligence updates in a stochastic model is critical for the defense analysis community. A robust intelligence updating algorithm has the propensity to make a stochastic model better. With better stochastic modeling, analysts are able to provide decision makers with better distributional information on probable whereabouts of targets of interest. With better information, decision makers are able to direct the appropriate assets in a more efficient manner to maximize mission effectiveness. The particle filter has the capability of providing a flexible method for incorporating intelligence updates.

5.1 Summary

Stochastic models are useful in a variety of different defense applications, from modeling combat to search and detection. An important element of stochastic models are their ability to incorporate additional information as it arrives. This is particularly true in time series models where the system state is being modeled over time. The ability to incorporate these updates is as important as the modeling itself. This thesis attempts to explore the relationship between a particular stochastic model and an information updating process in the context of maritime narcotics trafficking in the Eastern Pacific Ocean near South and Central America. A simulated Brownian bridge model is used as the stochastic model for modeling target movement and particle filtering is the information updating process for incorporating intelligence updates.

The simulated Brownian bridge model attempts to model target motion by simulating Brownian bridges. The Brownian bridge is a continuous time stochastic process that is Brownian motion fixed at two end points. The process starts at a specific location and ends at another location. A simulated Brownian motion path is generated between the two end points. With this underlying process, the simulated Brownian bridge model is specifically used to model movement behavior where there are defined areas of high activity for targets of interest, be it home ranges in animal movement (Bullard 1999) or target start and end locations in the South China Sea (Cheng 2016). This thesis and Cheng (2016) use

Brownian bridges to model the paths of targets of interest. These paths are aggregated to form a probabilistic heat map of target locations at predetermined time steps.

Using the SBBM as the underlying stochastic model, the next step is to incorporate an information updating procedure. The method that is implemented in this thesis is particle filtering. The benefit of using particle filtering is that there are minimal exogenous assumptions in its implementation, as opposed to parametric filtering methods like the Kalman filter. Particle filtering provides a robust method of information updating because there are no parametric restrictions. Monte Carlo sampling of the underlying probability density is used to create particles. At each iteration, these particles get a weight assignment based on the observed state through a predefined weighting scheme. The particles then get resampled with replacement based on the assigned weights. This causes particles with higher weights to have a higher probability of being carried over to the next iteration. There are some inherent issues associated with particle filtering, specifically particle degeneracy and the curse of dimensionality. This thesis develops a unique technique to manage particle degeneracy in the context of a Brownian bridge model. Particle degeneracy mitigation techniques such as roughening and adaptive resampling are implemented in working algorithms. As a consequence of particle degeneracy mitigation, the curse of dimensionality is also reduced. Lastly, this thesis offers an advanced particle filtering algorithm that is used in conjunction with the SBBM.

The next step is to incorporate a particle filter as the information updating method to the SBBM. The SBBM creates a series of Brownian bridge paths from defined start and end points. Sensors are designated by the user to gather intelligence updates. These sensors are placed along the projected target path, and provide the information updates to the model as the particles pass through during active times when the sensor is on. The particle filter incorporates the intelligence received by the sensors through a weighting scheme that uses sensor active times, particle locations in reference to the sensor, sensor probabilities of detection and a Boolean detection flag that differentiates between positive and negative sightings of the target. A roughening technique that creates new Brownian bridge paths at the point of resampling is implemented to mitigate particle degeneracy. An adaptive resampling technique that "activates" the particle filter in conjunction with sensor active times decreases the computational complexity of the model. This advanced model is compared against a basic version of the SBBM. Experimentation is conducted varying

weighting schemes, sensor areas, and resampling techniques.

A robust information updating process has the potential to increase the fidelity of the stochastic model significantly. This thesis shows that particle filtering is a robust and flexible method for incorporating intelligence updates to a SBBM. With this intelligence updating algorithm, this model is capable of serving as the stochastic model for an optimization that maximizes the overall probability of detection leading to a more efficient placement of sensors.

5.2   Future Work

With the particle filter implemented in the SBBM, a logical next step is to conduct experimentation against a SBBM with a Kalman filter as the information updating algorithm. A recommended method is to create multiple target tracks based on historical trends. The track can simulate a real narcotics smuggling route. Each of the stochastic models is then implemented using the series of routes as inputs. Additional factors such as target speed, sensor placement, and probability of detection, can be incorporated via an efficient design of experiments. A good measure of effectiveness (MOE) is the Brier Score or Logarithmic Score (which measure the reliability of probabilistic predictions) at each time step to determine which algorithm performs better.

Once the level of fidelity offered by the particle filter has been validated, another possible application for this model is in a stochastic optimization problem. This model can be applied to any target tracking scenario in a possibly different AOR. The SBBM can serve as the foundation model in this nonlinear programming problem where the decision variables are the location and type of sensor used. The enhanced SBBM with particle filtering provides the analyst with a robust method for determining overall probabilities of detection while employing a flexible weighting scheme.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Alm SE (2002) Simple random walk. Retrieved November 14, http://www2.math.uu.se/~sea/kurser/stokprocmn1/slumpvandring_eng.pdf.

Boyd S (2009) The Kalman filter. Retrieved November 9, http://stanford.edu/class/ee363/lectures/kf.pdf.

Buchin K, Sijben S, Arseneau T, Willems EP (2012) Detecting movement patterns using Brownian bridges. *Proc. of the 20th International Conf. on Advances in Geographic Information Systems* (Association for Computing Machinery, New York), 119–128.

Bullard F (1999) Estimating the home range of an animal: A Brownian bridge approach. Master's thesis, The University of North Carolina at Chapel Hill.

Campos S (2014) An analysis of mathematical models to improve counterdrug smuggling operations. Master's thesis, Naval Postgraduate School.

Cheng CC (2016) A Brownian bridge movement model to track mobile targets. Master's thesis, Naval Postgraduate School.

Daum F (2005) Nonlinear filters: Beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine* 20(8):57–69.

Daum F, Huang J (2003) Curse of dimensionality and particle filters. *Proc. 2003 Aerospace Conf.* (Institute of Electrical and Electronic Engineers, Washington, DC), 1979-1993.

Doucet A, de Freitas N, Gordon N (2001) An introduction to Sequential Monte Carlo methods. Doucet A, de Freitas, Gordon N, eds. *Sequential Monte Carlo Methods in Practice* (Springer, New York), 3–14.

Doucet A, Johansen AM (2009) A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering* 12(656-704):3.

Dudley SS (2010) Drug trafficking organizations in Central America: Transportistas, Mexican cartels and maras. Olsen E, Shirk D, Selee A, eds. *Shared Responsibility* (University of San Diego Trans-Border Institute, San Diego) 63–93.

Gordon NJ, Salmond DJ, Smith AF (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* 140(2):107–113.

Gustafsson F (2010) Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine* 25(7):53–82.

Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R, Nordlund PJ (2002) Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing* 50(2):425–437.

Henry (2011) Probability distribution of coverage of a set after "x" independently, randomly selected members of the set. Retrieved November 1, http://math.stackexchange.com/q/32816.

Horne JS, Garton EO, Krone SM, Lewis JS (2007) Analyzing animal movements using Brownian bridges. *Ecology* 88(9):2354–2363.

Kalman RE (1960) A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 82(1):35–45.

Kelly M (2012) 16 maps of drug flow into the United States. *Business Insider Online*, (July 8), http://www.businessinsider.com/16-maps-of-drug-flow-into-the-united-states-2012-7.

Lalley S (2013) Brownian motion. Retrieved November 16, http://galton.uchicago.edu/ lalley/Courses/313/BrownianMotionCurrent.pdf.

Minor B (2011) Particle filters. Retrieved January 23, http://www.eecs.wsu.edu/ cook/ml/presentations/bryan.pdf.

National Search and Rescue Committee (2000) United States national search and rescue supplement to the international aeronautical and maritime search and rescue manual. National Search and Rescue Committee, Washington, DC.

Office of the Chief of Naval Operations (1997) Navy search and rescue tactical information document. Office of the Chief of Naval Operations, Washington, DC.

Orlande H, Colaco M, Dulikravich G, Vianna F, da Silva W, da Fonseca H, Fudym O (2011) Tutorial 10 Kalman and particle filters. *Advanced Spring School: Thermal Measurements and Inverse Techniques 5* (Mesures en Thermiques et Techniques Inverses, Roscoff, FR), 1–39.

Siegrist K (2015) Standard Brownian motion. Retrieved November 17, http://www.math.uah.edu/stat/brown/Standard.html.

United States Senate Caucus on International Narcotics Control (2012) Preventing a security crisis in the Caribbean. United States Senate Caucus on International Narcotics Control, Washington, DC.

Weisstein EW (2016) Stirling number of the second kind. Retrieved October 27, http://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html.

Wolpert DM, Ghahramani Z (2000) Computational principles of movement neuroscience. *Nature Neuroscience* 3:1212–1217.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California